DISS. ETH No. 19884

Building-Blocks for Event-Based Vision Sensors

A dissertation submitted to the

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

RAPHAEL BERNER

MSc. Electrical Engineering and Information Technology, ETH

born 14.05.1980

citizen of
Schafisheim (Aargau)

accepted on the recommendation of

Prof. Tobias Delbrück, examiner
Prof. Dr. Rodney Douglas, coexaminer
Dr. Piotr Dudek, coexaminer
2011

Acknowledgements

First and foremost I would like to thank my supervisor Tobi Delbruck for his guidance, enthusiasm, valuable input and his uncomplicated ways. I also want to thank him for his patience, understanding and encouraging words when something did not work as planned. I really can not imagine a better supervisor...

I would also like to thank warmly Shih-Chii Liu for long and fruitful discussions, suggestions and help especially, but not only, during the design of the new word-serial AER circuits.

I thank Patrick Lichtsteiner for giving me a lot of guidance and inspiration, for help, valuable input and discussions not only about vision sensors and neuromorphic engineering. His continuous fascination of the field of asynchronous vision sensors and his insights of semiconductor device physics have been very helpful and inspiring.

I thank my co-supervisor and co-leader of the Institute of Neuroinformatics (INI) Rodney Douglas for providing a very nice working atmosphere and for asking interesting questions. His continuing support and interest for dynamic vision sensors are very motivating.

I would like to thank Piotr Dudek for serving as a co-referee during this thesis. The presentations about his work and discussions about neuromorphic engineering provided an interesting insight in the related research field of focal-plane vision processor chips.

I thank all the current and former members of the INI hardware group for interesting questions and stimulating discussions. I would especially like to acknowledge Yingxue Wang for encouraging me to look again at the LogSum color change detection circuit after I dismissed it due to a flaw in my reasonings.

I thank Shih-Chii, Christina, Patrick, Dennis and Emre for reading parts of my thesis and making suggestions how to improve it.

I thank Simone, Kathrin, Lottie, Frank and Dave from the INI administrative team for uncomplicated help and support during all possible kinds of issues.

I would like to thank all members of Dennis' friendly lunch group for meaningful and pointless discussions, lunch breaks are a lot of fun with you guys!

I thank all members of the Institute of Neuroinformatics for the nice and stimulating working environment.

I acknowledge Paul Merolla and John Arthur from Standford University who

developed the LogSpice simulation tools I used heavily during the development of the AER communication circuits.

I thank Ryad Benosman from the Université Pierre et Marie Curie in Paris. His enthusiasm for the dynamic vision sensors was and is very motivating and he was one of the driving forces behind our decision to combine dynamic and sustained pathways in the way we implemented it.

I would like to thank Gion-Pitschen for dragging me along to the class called 'Computation in Neuromorphic Analog VLSI' taught by Tobi Delbruck, Shih-Chii Liu and Giacomo Indiveri. This class proved to be a milestone for me.

During my studies and my PhD, my parents Werner and Annemarie Berner provided endless support, encouraging, understanding and love. I thank you very much! I would also like to thank my sister Cornelia, my brother Simon and their families.

Last but not least I would like to thank Christina for all the support and love she gave me during the last years. I would especially like to thank her for the patience she has shown during times of hard work just before deadlines and especially during the last few weeks of writing this thesis. This is dedicated to you!

This work was supported by the Swiss National Science Fund project 'Micropower Face and Voice detection' (project number 200021-112354), the EC 7th framework project 'SeeBetter' (project number 270324) and the Institute of Neuroinformatics.

Zurich, July 5, 2011

Raphael Berner

Abstract

Analyzing and understanding a visual scene and categorizing novel objects is so easy for us that we do not have to spend any effort to do it. Artificial vision systems on the other hand struggle in similar tasks. Despite the recent progress in computer vision and its increasing variety of applications, their functionality remains limited to well controlled environments solving very specific and well-defined tasks.

Biological vision systems are orders of magnitude more powerful and power-efficient than computer vision systems. Part of this efficiency comes from more efficient vision sensors. The mammalian retina (the light-sensing tissue at the back of the eye) is a highly developed neural circuit that preprocesses the visual input. The amount of sensory data sent to the brain is reduced by this preprocessing, and important visual features like local contrast, motion direction, etc. are extracted.

Computer vision systems rely on concepts and cameras that are optimized for recording picture sequences. These cameras and the entire processing chain of computer vision systems rely on the concept of 'frames'. A frame is a matrix of values representing the intensities at every spatial location. The camera sends frames at a fixed rate to the processing chain. Frame-based cameras have been developed and optimized for several decades and allow very small pixels and thus high resolution. But using frames has several disadvantages. The fixed frame rate limits the temporal resolution of the system. Each pixel of every frame must be read and processed, even if its value did not change. This repeated processing of redundant data is highly inefficient.

The research domain of neuromorphic engineering uses concepts found in biological systems to build devices that are more efficient than conventional devices. During the last twenty years neuromorphic engineers have built vision sensors that differ fundamentally from traditional cameras. However most of these devices have low resolution and suffer from strong fixed-pattern noise and low usability. Hence they are not suited to be employed in real-world applications.

At the Institute of Neuroinformatics, Patrick Lichtsteiner and Tobi Delbruck developed the dynamic vision sensor, a neuromorphic vision sensor that proved to be useful in very different applications. In this device, the pixels emit asynchronous events that indicate a relative change of intensity with very low latency. This kind of output completely suppresses temporal redundancy. Thanks to the reduced data rate, less computing power is necessary to process the data; and the low latency

allows fast reaction times for real-time vision systems, which is useful for example in mobile robotics.

This thesis addresses three limitations of the dynamic vision sensor:

- The dynamic vision sensor is, like all other neuromorphic vision sensors presented so far, only monochromatic.
- If the dynamic vision sensor is fixed, it is not able to see static objects.
- Despite the event-driven output, the communication bandwidth of the dynamic vision sensor is insufficient for busy scenes and limits the scalability of the current design.

This thesis presents a novel pixel circuit which detects changes in wavelength and asynchronously emits events indicating whether the mean wavelength has decreased (BLUER) or increased (REDDER). The color sensitivity is achieved by exploiting the basic property of silicon that the absorption depth of photons is wavelength-dependent. Thus no special process steps are needed for the fabrication of our pixel. The pixel uses a buried double junction, which is a stack of two photodiodes at different depths from the silicon surface. Due to the different depth and the fact that photons with shorter wavelengths are absorbed closer to the semiconductor surface, the two photodiodes of the buried double junction have different spectral sensitivities. Our pixel circuit uses two logarithmic current-to-voltage converters and a summing amplifier to calculate the ratio of the two junction currents. The wavelength information is contained in this ratio.

Single test pixels achieve encouraging color sensitivity, but the color sensitivity of our small pixel array is limited due to mismatch. Further research is necessary to build a neuromorphic color vision sensor that is usable in real-world applications.

The second achievement of this thesis is a compact combination of a dynamic pixel circuit with a readout of the intensity, using only a single photodiode for both dynamic and sustained pathways. The circuit uses a MOSFET transistor running in subthreshold to logarithmically convert the photocurrent to a voltage. The pixel array uses on-chip calibration circuits to suppress fixed-pattern noise.

The intensity readout is functional but signal quality is heavily degraded by the dynamic pathway emitting events. Contrast sensitivity is low due to low signal swing and incomplete suppression of fixed-pattern noise at low to intermediate light intensities. Circuit improvements are proposed to address these issues.

The third achievement of this thesis are new asynchronous communication circuits that improve the communication bandwidth of the dynamic vision sensor by more than an order of magnitude. This allows vision sensors with higher resolution without sacrificing per-pixel bandwidth.

With the presented compact pixel circuit and the new communication circuits, this thesis provides important steps towards more powerful neuromorphic vision sensors.

Zusammenfassung

Visuelle Szenen zu analysieren und darin enthaltene Objekte zu kategorisieren gelingt uns Menschen mühelos. Rechnergestützte Bildanalyseverfahren hingegen haben trotz grosser Fortschritte in den letzten Jahren Mühe mit vergleichbaren Aufgaben. Bildanalyseverfahren werden vielfältig und mit grossem Erfolg eingesetzt, sind aber beschränkt auf gut ausgeleuchtete Szenen und das Erledigen von genau definierten Aufgaben.

Die höhere Effizienz von biologischen Systemen ist zum Teil auf bessere und effizientere Sensoren zurückzuführen. Die Sehrinde der Säugetiere ist ein hochentwickeltes neuronales Netzwerk, welches die visuellen Eingangssignale vorverarbeitet. Diese Vorverarbeitung extrahiert wichtige visuelle Merkmale wie den lokalen Kontrast und Bewegungsinformationen. Die Menge der ans Hirn gesendeten Informationen wird durch diese Vorverarbeitung reduziert.

Die von rechnergestützten Bildanalyseverfahren verwendeten Kameras unterscheiden sich fundamental von den Augen biologischer Lebewesen. Videokameras benutzen das Konzept von "Frames" (englisch für Einzelbild). Ein Frame ist eine Matrix von Werten, welche die Lichtintensität von jedem Pixel zu einem bestimmten Zeitpunkt repräsentieren. Die Kamera schickt die Frames in regelmässigen Zeitabständen zur Bildverarbeitungskette. Frame-basierte Kameras werden seit mehreren Jahrzehnten weiterentwickelt und erlauben sehr kleine Pixel und damit hohe Auflösungen. Die Frame-basierte Architektur hat jedoch einige Nachteile. Die zeitliche Auflösung ist limitiert durch die Frame-Rate. In jedem Frame muss jedes Pixel ausgelesen und verarbeitet werden, auch wenn sich der Wert des Pixels nicht verändert hat. Dieses repetitive Verarbeiten redundanter Informationen ist ineffizient.

Das Forschungsgebiet "Neuromorphic Engineering" übernimmt Konzepte von neurologischen Systemen, um Geräte zu konstruieren, welche effizienter arbeiten als es mit konventionellen Ansätzen möglich wäre. In den letzten 20 Jahren wurden verschiedenste neuromorphe visuelle Sensoren vorgestellt, welche sich fundamental von konventionellen Kameras unterscheiden. Diese Sensoren versuchen einige Aspekte der Sehrinde zu implementieren. Allerdings sind die meisten dieser Sensoren ungeeignet für den alltäglichen Einsatz, da sie ein starkes feststehendes Störmuster, eine niedrige Auflösung sowie eine geringe Benutzerfreundlichkeit aufweisen.

Patrick Lichtsteiner und Tobias Delbrück haben am Institut für Neuroinformatik den "Dynamic Vision Sensor" entwickelt, welcher beweist, dass neuromorphe Sensoren in diversen Anwendungen sehr nützlich sein können. Die Pixel in diesem

Sensor detektieren relative Änderungen der Lichtintensität und senden asynchrone "ON-Events" und "OFF-Events" mit sehr geringer Latenz, je nach dem ob die Lichtintensität zugenommen oder abgenommen hat. Durch diese Art von Vorverarbeitung wird die zeitliche Redundanz eliminiert und die Menge an zu verarbeitenden Daten reduziert. Dies ermöglicht eine sehr schnelle Bildverarbeitung für Echtzeit-Systeme und die geringe Latenz erlaubt schnelle Reaktionszeiten zum Beispiel für Roboter.

Diese Doktorarbeit befasst sich mit drei Nachteilen des "Dynamic Vision Sensor":

- Der "Dynamic Vision Sensor" ist, wie alle neuromorphen visuellen Sensoren, nur monochrom.
- Wenn der "Dynamic Vision Sensor" fix montiert ist, können statische Objekte nicht wahrgenommen werden.
- Die Kommunikations-Bandbreite des "Dynamic Vision Sensor" ist nicht ausreichend für Szenen mit grossen Veränderungen. Diese Limitierung ist problematisch, wenn die Auflösung des "Dynamic Vision Sensor" erhöht werden soll.

Während dieser Doktorarbeit haben wir ein Pixel entwickelt, welches Änderungen von Licht-Wellenlängen wahrnehmen kann. Es sendet asynchrone "REDDER-Events", wenn sich die Wellenlänge des einfallenden Lichts erhöht, und "BLUER-Events", wenn sich die Wellenlänge des einfallenden Lichts vermindert. Zur Realisierung der Farbsensitivität nutzt das Pixel die intrinsische Eigenschaft von Silizium, dass Photonen mit kurzer Wellenlänge näher an der Oberfläche absorbiert werden als Photonen mit grosser Wellenlänge. Das Pixel nutzt eine vertikale Anordnung von zwei p-n Übergängen, welche in jeder normalen CMOS-Technologie verfügbar ist. Da sich die Übergänge in unterschiedlicher Distanz zur Oberfläche befinden, ändert sich das Verhältnis zwischen beiden Photoströmen mit der Wellenlänge. Unser Pixel benutzt logarithmische Strom-zu-Spannungs-Wandler und einen Summenverstärker zur Berechnung der Differenz der Logarithmen der zwei Photoströme. Die Differenz der Logarithmen ist äquivalent zum Logarithmus des Verhältnisses und kodiert deshalb die mittlere Wellenlänge des einfallenden Lichts.

Einzelne Testpixel erreichen ermutigende Farbsensitivität, aber die Farbsensitivität unseres kleinen visuellen Sensors mit 32×32 Pixeln ist limitiert durch Diskrepanzen zwischen den Pixeln. Um einen brauchbaren neuromorphen Farbsensor zu entwickeln ist daher weitere Forschungsarbeit nötig.

Die zweite Errungenschaft dieser Doktorarbeit ist ein kompaktes Pixel, bei welchem neben der asynchronen Übermittlung von Änderungen auch die Intensität ausgelesen werden kann. Das Pixel nutzt nur eine Photodiode für beide Ausgangsgrössen. Ein Transistor im Unterschwellen-Bereich konvertiert den Photostrom logarithmisch in eine Spannung, welche dann ausgelesen wird. Der Sensor nutzt eine Kalibrierungs-Schaltung zur Unterdrückung von festen Störmustern.

Wir konnten zeigen, dass das Auslesen der Intensitätswerte funktioniert. Allerdings wird die Bildqualität durch den asynchronen Ausgangs-Pfad der Pixel beeinträchtig. Die Kontrastsensitivität ist nur zufriedenstellend für hohe Lichtintensitäten, bei tiefen Lichtintensitäten ist die Unterdrückung von festen Störmustern nicht ausreichend. Wir präsentieren mögliche Verbesserungen der Schaltung um diese Probleme zu beheben.

Die dritte Errungenschaft dieser Doktorarbeit ist eine neue asynchrone Kommunikations-Schaltung, welche die Kommunikations-Bandbreite des "Dynamic Vision Sensor" um mehr als eine Dekade erhöht. Dies wird uns in Zukunft ermöglichen, visuelle Sensoren mit höherer Auflösung zu entwickeln.

Mit der kompakten Pixel-Schaltung und der neue Kommunikations-Schaltung präsentiert diese Doktorarbeit wichtige Schritte in Richtung leistungsfähiger neuromorpher visueller Sensoren.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biological vision systems are very different from conventional computer vision systems and much more efficient. Part of this increased efficiency comes from more efficient vision sensors. In recent years, several research groups around the world presented vision sensors which are inspired from biology to try to improve the efficiency of artificial vision systems. All of these bio-inspired vision sensors are monochrome, yet color information is used by a lot of animals. This thesis addresses this topic and investigates circuits for building bio-inspired asynchronous event-based color vision sensors. A short summary of the achievements is presented in Section 1.7.

1.1 Neuromorphic Engineering

Even though modern computers have become exponentionally faster and more powerful during the last decades, they are still outperformed by comparatively simple biological brains in most aspects of real-world interaction, such as object recognition, sound recognition, autonomous locomotion and navigating in challenging environments. Despite having a brain the size of a sesame seed containing about one million neurons, a bee controls its flight, navigates and also communicates with peers [1]. Artificial micro-flyers are far away from achieving comparable performance and autonomously flying and navigating vehicles need computers that weigh in the order of kilograms and use tens of watts for solving similar tasks.

The underlying computational architectures of biological systems and computers are very different. Computers use synchronized digital logic, a central processing unit (CPU) and memory "far away" from the CPU. Computing is traditionally done in a sequential way. Brains however combine analog and digital computation, memory and processing are not separable and computation is done massively parallel with comparatively slow and simple units which are highly connected.

The pioneer of neuromorphic engineering, Carver Mead, argued that the underlying device physics of CMOS technology and the neural substrate is not fundamentally different, as they both rely mainly on Boltzmann statistics. He said

that therefore it should be possible to implement biologically inspired devices that come closer to the performance shown by biological systems than current artificial systems [2, 3].

"Neuromorphic Engineering" is the name of the research field which tries to morph concepts learned from biology into engineered systems. There are different motivations for neuromorphic engineering:

- Implementing models presented by neuro-scientists without direct applications to try to advance understanding of neural computation [4, 5, 6]. Or to put it in Carver Mead's words [7]:

  > If we *really* understand a system, we will be able to build it. Conversely, we can be sure that a system is not fully understood until a working model has been synthesized and successfully demonstrated.

- Building tools for neuro-scientists, for example large-scale neural network simulation hardware like the hardware developed during the FACETS and BrainScaleS projects [8, 9, 10], the NeuroGrid project [11] or the SpiNNaker project [12, 13] or model neural systems for teaching [14].

- Building interfaces to biological systems like decoders for multi-electrode arrays [15] and eventually neuro-prostheses [16]. For example Vogelstein et al. presented a study where they used a central pattern generator built from silicon neurons to control locomotion of an anesthetized cat [17].

- Implementing basic principles and taking inspiration from neuro-science without faithful mimicking biological circuits to build engineered devices for practical applications [18, 19, 20, 21].

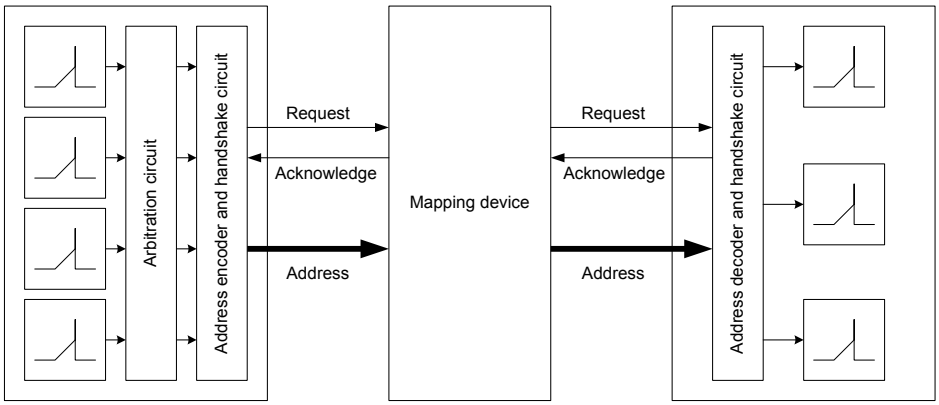
Twenty years of research in neuromorphic engineering led to very different types of devices, from many different silicon neuron chips [22] to silicon retinas [7], olfaction chips [23], silicon cochleae [24, 25], and so on.

This thesis focuses on neuromorphic vision sensor circuits that take inspiration from biology, but do not try to copy or mimic any specific biological circuits.

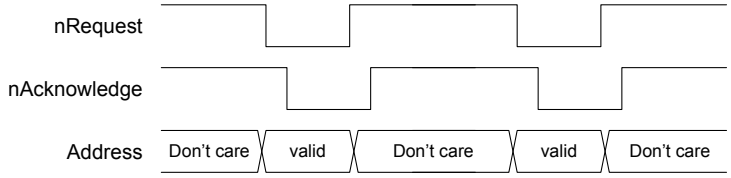
1.1.1 Address Event Representation

Even though there might be similarities in the underlying device physics of silicon technology and biological brains, there are also fundamental differences. Brains evolve during a lifetime and can grow new connections and remove connections between neurons, silicon devices however are fixed structures and metal connections can not be changed.

Brains have a very high connectivity; a fanout of several thousand is very common for neurons in the cortex. Because of the limited number of metal layers in CMOS technology, it is not possible to implement a similar connectivity using dedicated wires.



(a) AER concept. The sender encodes the address of the firing neuron, an optional mapping device can modify the address and the decoder of the receiving device sends the spike to the correct neuron.



(b) Example time-line of the AER signals. The request and acknowledge signals are usually active low. The address must be valid during the time the request is active.

Figure 1.1: Address-Event-Representation concept

Silicon technology on the other hand can operate six to seven orders of magnitudes faster than biology's ionic devices. This allows time-multiplexing neuron-to-neuron connections on a single communication channel instead of using dedicated wires. This further allows dynamically changing connections between devices by changing the input-output mapping of this communication channel with dedicated mapping devices.

The AER (*A*ddress *E*vent *R*epresentation) is an asynchronous communication protocol used for communication between AER building blocks. It has been proposed in Carver Mead's lab [26] to implement connectivity between neuromorphic circuits.

Several version of the protocol have been proposed, ranging from point-to-point connection to multiple sender and multiple receiver variants. In this work we use single sender/single receiver communication.

The AER single sender/single receiver protocol usually uses a four-phase handshake with a request line driven by the sender, an acknowledge line driven by the receiver and a set of address lines driven by the sender. Fig. 1.1 illustrates the concept of AER and shows example time-lines of the signals.

The CAVIAR project [27] demonstrated one of the largest AER multi-chip systems so far, implementing a feed-forward vision systems across many neuromorphic chips with a total of over 45'000 neurons. During this project, several useful AER tools were developed [28, 29, 30].

The AER protocol is continuously extended and improved. Boahen's group proposed a word serial AER protocol which achieves an increase in address space and throughput [31, 32, 33] and recently also proposed an AER tree structure [34]. Several groups are working on serial AER communication, either using off-the-shelf components [35, 36, 37] or developing on-chip serial AER interfaces [38].

1.2 Biological Vision

In this section, a short and very simplified overview of the human visual system is given. The human visual system shows amazing capabilities. We are able to see and detect objects in very dark and very bright conditions. We can detect and distinguish a vast amount of object classes, even in very difficult lighting conditions and when the object is partly occluded. We are able to estimate the distance to objects and their movements, and so on. The human visual system can solve most of these tasks orders of magnitude better and more power efficiently than any artificial vision system.

1.2.1 Visual pathway

Fig. 1.2 shows a simplified sketch of the human visual pathway. The eyes capture visual information and perform highly parallel preprocessing and data reduction. The extracted information is relayed via the optic chiasm to the lateral geniculate nucleus (LGN). The optic chiasm collects information from both eyes and sends information from the left visual field to the right brain hemisphere and vice versa. Additional to the visual input from the optic chiasm, the LGN also receives feedback input from higher visual areas that appears to provide information about attention and expectations and modulates the processing in the LGN. The output of the LGN is sent to the primary visual cortex and from there to higher visual areas.

Vastly simplified, the visual systems is divided in two pathways, one of which is mostly transient and monochrome, and responsible for detecting object locations, boundaries and motion. The other pathway is color sensitive, provides fine details of objects and dominates input to object classification and recognition areas.

1.2.2 The eye

Fig. 1.3 shows a schematic cross section of the human eye. Light enters through the cornea, passes through the aqueous humor and the iris and is then focused by the lens onto the retinal layer at the back of the eye. The shape of the lens can be changed to adjust the focal point to the distance of the object the eye is looking at.

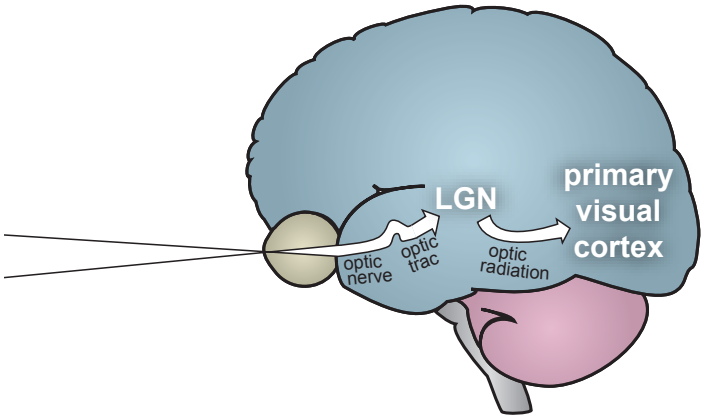


Figure 1.2: Visual pathway from the retina to the visual cortex, reprinted with permission from [39].

To cope with the big range of light intensities found in natural conditions, the eye can sense a dynamic range of more than 8 decades with a contrast sensitivity of one to two percent for good illumination [40].

1.2.3 The retina

The tissue at the back of the eye that converts the incoming light into signals the brain can understand is called the retina. It consist of several layers of neurons, an abstraction is shown in Fig. 1.4.

At the back of the retina lie the two different kinds of photoreceptors, the rods and the cones. Rods are monochrome and very light sensitive, they can respond to single photons and thus enable night vision. The cones on the other hand need more light to function. There are three different types of cone photoreceptors sensitive to different parts of the wavelength spectrum. The cones thus enable us the perception of colors. The entire retina contains about 7 million cones and 75 to 150 million rods.

The photoreceptors are coupled to each other using gap junctions and are connected to horizontal cells and bipolar cells. Horizontal, bipolar and amacarine cells are highly connected and perform spatial and temporal filtering. The output of the retina is formed by the ganglion cells. There are at least 27 different kinds of ganglion cells coding different kinds of visual features [41], but not all these ganglion cell types are understood. The ganglion cells code visual features, which means the output of the retina is data driven. Simply put, if a ganglion cells fires, it means a certain feature is present.

In the center of the retina, just opposite the lens lies the fovea, an area very densely packed with cone photoreceptors. The fovea allows high angular resolution and color vision. Here the mapping of photoreceptors to ganglion cells is almost one-to-one.

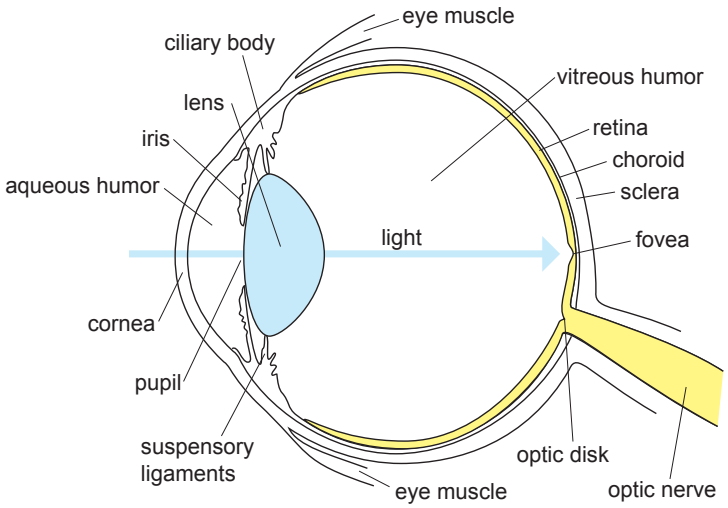


Figure 1.3: Schematic drawing of the eye, reprinted with permission from [39].

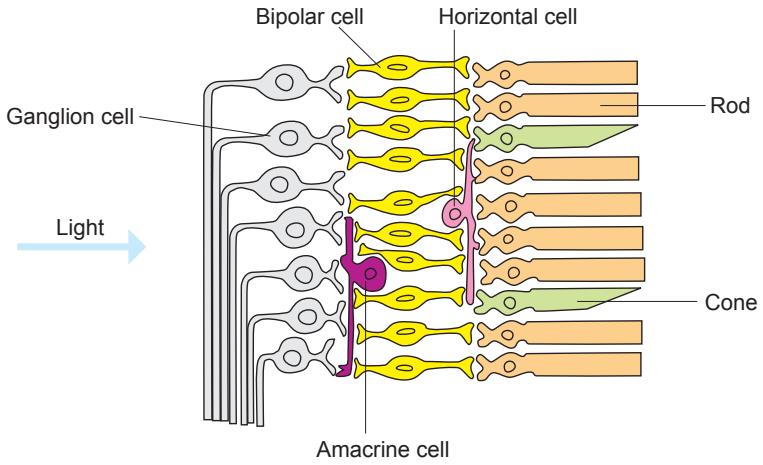


Figure 1.4: Schematic drawing of the neural layers in the retina, reprinted with permission from [39].

Outside the fovea, the density of rod photoreceptors is higher than the density of cone photoreceptors, and the mapping from photoreceptors to ganglion cells is many to one. The farther away from the fovea, receptive fields of ganglion cells get increasingly bigger and thus visual acuity increasingly lower.

In summary, the retina is not just a simple light sensing tissue, it does not send snapshots of the visual scene to the brain. The retina performs feature extraction and data reduction to lower the processing burden of higher visual areas. The retina is a very complex network of many different cell types still not very well understood and thus is the focus of ongoing research. At this point, faithful mimicking of the sensing and all the retinal processing is not conceivable in a single silicon device. However, neuromorphic engineering can take inspiration and use some concepts like data-driven output, adaptation and reduction of temporal and spatial redundancy.

1.3 Traditional Approach to Computer Vision: Frames

Computer vision (also called machine vision) is the discipline which creates algorithms to extract information of visual input data, usually pictures or movies. This includes optical inspection in industrial applications, autonomous vehicles, object or people tracking and counting, automatic surveillance etc.

Computer vision uses cameras and concepts that originally were built for recording and playback of visual scenes to human observers. These cameras have been optimized over decades for recording pleasing pictures, therefore they are not optimized for doing computer vision.

Almost all imagers, cameras and devices for computer vision, movie recording and playback are built around the concept of frames. A frame represents a time-averaged snapshot of a visual scene. Movies are sequences of frames with a fixed frame rate, which is usually around 25 frames per second, but can also go as high as 10 000 frames per second for special applications. Any computation in the time-domain, for example tracking of an object, is done on this sequence of snapshots. This can lead to the so-called frame-correspondence problem. This term describes the problem that it is not always obvious to decide which object in one frame corresponds to which object in the subsequent frame. This problem is especially severe if multiple fast moving similar objects are present.

1.3.1 Advantages and disadvantages of the frame-based approach

The frame based approach has undeniable advantages:

- Cameras and devices have been evolved over decades and are well understood.
- Frame-based imagers can have very small pixels (a $1.1 \times 1.1 \mu m^2$ pixel has been reported at the 2011 international image sensor workshop [42]). That allows high resolution cameras in small packages at low cost. The latest compact cameras and even cell phones can record still pictures with a resolution of up

to 14 mega-pixels and movie clips in high-definition resolution (1920×1080). High resolution is very nice for large print-outs and cinematic movie playback, but for computer vision more resolution means more computational power is necessary to process the additional pixels.

- The concept of frames is easy to understand and algorithms to process frame-based data have been the focus of computer vision research for several decades and are thus highly evolved.

In the computer vision community, frames are taken for granted. But the fixed frame rate of traditional vision system imposes several disadvantages:

- Temporal resolution is limited, thus anything happening faster than the frame rate will be aliased to the frame rate. Increasing the frame rate increases the amount of data to be processed.

- Related to the limited temporal resolution is the frame-correspondence problem for time-domain computation.

- Most of the output data in frame-based cameras is redundant. In every frame, all pixels are read-out, even if the light falling on a pixel did not change and therefore this pixel provides no new information.

- Usually the dynamic range of traditional cameras is limited to at most 70dB, because integration time, gain and pixel capacity are the same for all the pixels. Usually low dynamic range is dealt with by expensive artificial lighting.

- Relatively large system level power consumption because all pixels have to be converted to digital values and all the data has to be transmitted and processed at a high rate.

1.3.2 CCD and CMOS imagers

In 2009, Willard Boyle and George Smith received the Nobel prize for Physics for their invention of the charge-coupled device (CCD) in 1969. This type of image sensor has dominated digital imaging for several decades.

Eric Fossum predicted in 1993 that CMOS imagers will eventually replace CCDs [43] because the CMOS technology allows to put ADC circuits or processing elements on the same die [44]. In the last few years his prediction has come to reality, CMOS sensors using active pixel sensors (APS, see section 2.1 for a detailed explanation) have outnumbered the use of CCD sensors in consumer cameras. CCDs are still used in medium format cameras and special applications like astronomical imaging.

Consumer devices with CMOS sensors exclusively use the APS circuit, but research is ongoing to improve and evolve this pixel sensor for example to increase dynamic range [45] or reduce read-out noise [46].

1.4 Neuromorphic vision sensors

Neuromorphic vision sensors differ from conventional cameras; they try to address one or several of the shortcomings listed above. Many different kinds of neuromorphic vision sensors have been published during more than two decades of neuromorphic engineering. Many of them were special purpose vision sensors like optical flow chips, motion extraction sensors, etc. [47, 48, 49, 50]. Here we focus on general-purpose vision sensors and describe the most important chips presented in the last decade.

Carver Mead and Misha Mahowald pioneered the field of neuromorphic vision, they presented the first silicon retinas [7, 51, 52]. These devices implemented parts of the spatial preprocessing done in biological retinas, but did not yet have an asynchronous data-driven output. These early sensors have interesting properties, but suffered from strong fixed-pattern noise which made them unusable for real-word applications.

The first silicon retinas tried to mimic as much as possible the processing done by the first layers of the mammalian retina. This needed complex circuits and usually yielded relatively bad performance due to mismatch in the sub-threshold current-mode processing. Most of the more recently published neuromorphic vision sensors are more abstracted and focus on one or two aspects of the retinal preprocessing. An exception is the sensor presented by Zaghloul and Boahen [53] that incorporates both sustained and transient pathways with temporal and spatial filtering and event-based output. It is the sensor that models closest the properties of the early visual processing in the mammalian retina. However, this device again suffers from low dynamic range and very strong mismatch; firing rates between pixels vary up to one or two decades.

1.4.1 Asynchronous image sensors

In the image sensor presented by Culurciello et al. in 2003, spiking frequency in each pixel directly codes pixel illumination with a linear relationship [54]. In 2008, Olsson and Haefliger presented a similar device with much improved fixed pattern noise [55]. Both these devices achieve high dynamic range, but regions with low illumination are communicated with much higher latency than regions with high illumination. Very bright regions can saturate the bus.

These problems are addressed with time-to-first spike imagers [56, 57, 58, 59]. This type of sensor is frame-based, but instead of integrating over a fixed exposure time as in the conventional APS pixel [43] (see section 2.1), the integration is over a fixed voltage. In each frame, the brightest pixels spike first. The integration voltage range can also be varied during a frame to ensure that every pixel is spiking in the given time window. These sensors have very high dynamic range but uniform scenes cause event collisions and timing errors and thus errors in the scene representation.

Conceptually these image sensors differ very little from conventional image sensors, all they do is convert the light intensities falling onto the pixels to some

electrical signal, without doing any preprocessing. The sensors presented in the following subsections are more interesting from a vision processing point of view, because they compute some more advanced properties of the visual scene or do data reduction. Specifications of several of the presented sensors are compared in table 1.1.

1.4.2 Spatial contrast sensors

The silicon retina presented by Kameda and Yagi in 2003 [60] mimics the Laplacian-Gaussian-like spatial preprocessing seen in biological retinas, similar to earlier silicon retinas [7]. In contrast to the time-continuous logarithmic front-end used in the Mead-Mahowald silicon retina, Kameda's sensor uses an APS pixel front-end to achieve high input signal swing and offset-compensation circuits to achieve higher sensitivity and lower fixed-pattern noise. However, the pixels are complex and thus fill-factor is low in this sensor. The sensor is also capable of emulating a transient pathway by means of on-chip frame-differencing. The sensor is not data driven; all the pixels are scanned and the amount of data to be transmitted and processed is not reduced. In 2005, this group presented an improved version of this sensor with increased resolution [61] and recently with a logarithmic front-end based on stepped reset voltage [62].

A group at the CSEM in Neuchâtel presented a spatial contrast sensor in 2003 [63]. This sensor is frame based similar to the time-to-first-spike imagers, but the asynchronous output events do not code pixel intensities but spatial nearest neighbor contrast. In each frame, the pixels are transmitted in decreasing order of spatial contrast, which means highest spatial contrast cues are transmitted first. Edges are thinned by non-maximum-suppression circuits and therefore the amount of data to be transmitted and processed is reduced.

In 2010, Leñero-Bardallo et al. [64] presented an asynchronous spiking spatial contrast vision sensor that improved on previous work by Boahen and Andreou [65]. The sensor presented by Boahen and Andreou uses a compact current-mode CMOS circuit to model antagonistic center/surround receptive fields, however it had low dynamic range, the receptive field changed with illumination level and it suffered from mismatch. To reduce mismatch, Leñero-Bardallo et al. used calibration circuits in each pixel and they improved the biasing circuits to reduce the receptive field size dependence on the illumination level. The circuits asynchronously output events which indicate signed spatial contrast and have very low mismatch despite relying on current-mode sub-threshold circuits. The sensor can also be configured to operate in a Time-to-First-Spike contrast computation mode. However, pixels are complex and fill factor is low.

1.4.3 Temporal difference and temporal contrast change detection

To boost data transmission efficiency and lower the computational burden on processing stages, a class of sensors suppresses temporal redundancy right at the pixel

	Shimonomura et al. [62]	Ruedi et al. [63]	Leñero et al. [64]	DVS128 [66]	ATIS [67]	ASAP2 [68]
Functionality	SC, Scan, FR	SC, FR	SC, EV	TC, EV	TC, EV, INT	SIMD general purpose digital vision chip
Process Technology	$0.25\mu m$ 3M1P	$0.5\mu m$ 3M2P	$0.35\mu m$ 4M2P	$0.35\mu m$ 4M2P	$180nm$ 6M1P	$180nm$ 6M1P
Pixel size μm^2	87×87	69×69	81.5×76.5	40×40	30×30	51×54
Fill factor	8.5%	9%	2%	8.1%	10-20%	12.3%
Array size	64×64	128×128	32×32	128×128	304×240	80×80
Pixel Complexity	63T	?	131 T 2 C	26 T 3 C	77 T, 3C 2 PD	588 T
Dynamic range	100dB	120dB	100dB	120dB	125dB	?
FPN	?	<2%	0.9 %	2.1%	<0.25% (INT) ? (TC)	?
Power consumption mW	32	300	0.66-6.6	24	50-175	41

Table 1.1: Neuromorphic vision sensor comparison. Legend: SC: spatial contrast; TC: temporal contrast; FR: frame based; EV: event-based output; Scan: scanned output; INT: intensity data output. T: Transistor, C: capacitor, PD: photodiode

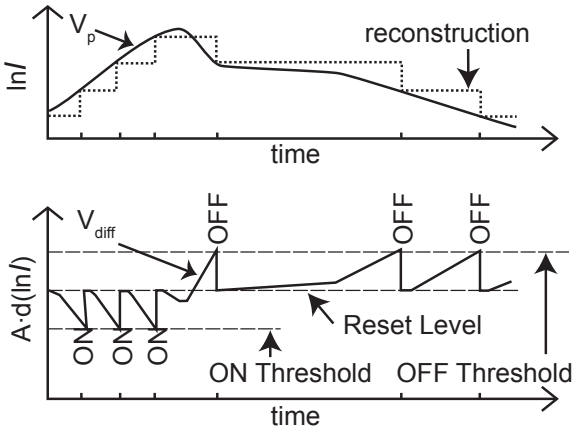


Figure 1.5: Temporal derivative vision sensor principle, reprinted with permission from [39].

level. Only pixels with new information transmit data.

In 2005 Mallik et al. [69] reported a temporal change threshold detection imager. This frame-based sensor with APS pixel front-end detects quantized absolute change in illumination. The sensor has small pixels but limited dynamic range. Absolute illumination change detection has the disadvantage that the same scene with different illumination responds differently (e.g. if half the scene is cast in shadow, the absolute changes in the shadow area are smaller because absolute illumination is smaller).

Temporal contrast detection sensors on the contrary report relative changes in pixel illumination. This has the advantage that it is inherently more robust to spatially varying scene illumination, because relative changes do not depend on global illumination. The principle is illustrated in Fig. 1.5.

The first temporal contrast silicon retina was developed by Jörg Kramer [70, 71]. This sensor proved that temporal contrast detection is useful, but had problems with self-sustained oscillations in the photoreceptor and relatively low sensitivity due to mismatch. This device generated events in pixels with threshold-exceeding values of temporal contrast. Slowly moving objects could escape detection.

The DVS128 dynamic vision sensor [66] presented by Lichtsteiner et al. improves on Kramer's work with a better pixel design that reduces photodiode capacitance, improves symmetry of the response, sensitivity and robustness against fabrication variations. Because the DVS128 is the basis of most of the work presented in this thesis, it is described in more detail in section 1.5.

In 2010, Leñero-Bardallo et al. presented a vision sensor that, compared to the DVS128, reduces latency, pixel area and improves contrast sensitivity at the expense of reduced intra-scene dynamic range, slightly increased pixel threshold variation and increased power consumption [72]. The pixel uses a different front-end with

reduced Miller-capacitance and therefore achieves higher bandwidth. It also uses an additional amplifier stage to increase contrast sensitivity. The authors report a very low latency of $3.6\mu s$ in bright lighting, enabling the sensor to see an object rotating at 10 000 revolutions per second. Contrast sensitivity is only slightly improved to 10%, compared to 15% in the DVS128. The pixel area is reduced because this pixel needs less capacitance compared to the DVS128, where almost 50 % of the pixel area is used by capacitors. However, manufactured in a modern process technology with metal-insulator-metal capacitors that can be stacked above the circuits, the pixel presented by Leñero-Bardallo et al. will need more area than Lichtsteiner's pixel.

In 2010, Posch et al. presented the ATIS (Asynchronous Time-based Image Sensor) [67]. This sensor combines in each pixel two complementary sensor circuits: a circuit equal to the DVS128 pixel that detects temporal contrast changes and a circuit that measures pixel illumination with a intensity-to-time conversion, similar to the time-to-first spike imagers. In contrast to time-to-first-spike imagers, the intensity measurement circuit of the ATIS is triggered by a temporal contrast change event, which means new intensity values are only transmitted after a change has happened, achieving real-time data compression. The intensity measurement circuits uses time-domain correlated double sampling (TCDS) to reduce fixed-pattern noise. The drawbacks of the ATIS are large area requirements because each pixel includes two photo-diodes and lots of circuitry, and also relatively high communication bandwidth requirements, because a change in a pixel leads to three events being transmitted: one to communicate the temporal contrast change and two to communicate the intensity value with TCDS. Nevertheless, this sensor achieves impressive performance and combines asynchronous change detection with imaging capabilities in a useful way.

1.4.4 Focal-plane pixel-parallel Vision Processor Chips

Pixel-parallel focal-plane vision processor chips are not directly neurally inspired, but nevertheless share some properties with biological systems. Processing is massively parallel with relatively simple processing units and memory sits close to every processing unit. However, there are important differences. Focal-plane pixel-parallel vision processor chips are, unlike biological systems, clocked systems executing a program defined by a series of instructions that have to be applied in the right order from some separate memory component. The algorithm is not implicitly defined by the circuits and the connections.

A group at the University of Manchester has built an impressive series of pixel-parallel vision processor chips [73, 74, 75]. These chips have in each pixel a photodiode, a set of analog registers and a switched-current analog processing unit that handles the basic operations addition, inversion and division by a constant. The pixels have connections to their neighbors.

The array is a SIMD (Single Instruction Multiple Data) processor, i.e. each pixel processing unit executes the same instruction. In the last version built in $0.35\mu m$

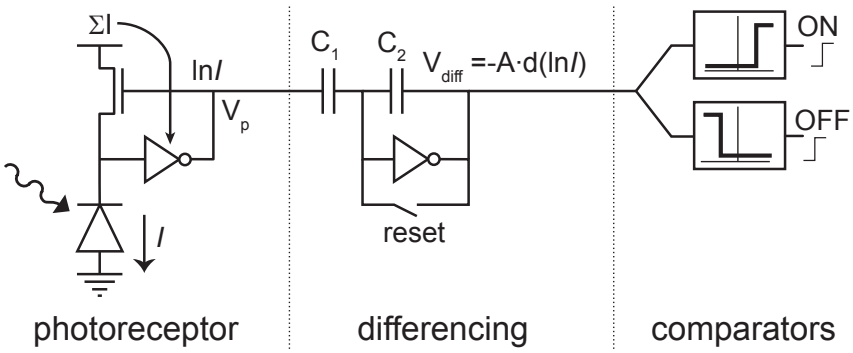


Figure 1.6: DVS128 pixel circuit abstraction, reprinted with permission from [39].

technology, the size of each complete pixel is $49.35 \times 49.35 \mu m^2$. This 128×128 array reaches 20 GIPS (giga-instructions per second) at a power consumption of 250mW, which compares favorably with conventional digital signal processors.

Recently, this group moved away from analog processing units. They presented an 80×80 general purpose digital vision chip [68] with a digital processing unit per pixel, because according to the authors robustness and stronger noise immunity make the digital approach more favorable in nanoscale CMOS technologies. The chip uses synchronous address-event read-out and has the ability to compute different types of events.

1.5 The DVS128 Temporal Contrast Vision Sensor

Most of the work presented in this thesis is based on the DVS128 silicon retina [39, 66] that was developed in the course of the CAVIAR project [27] by Patrick Lichtsteiner and Tobi Delbruck.

The DVS128 is a 128×128 array of independent, asynchronous pixels that communicate using AER circuits presented by Boahen [76]. The device asynchronously emits address-events indicating changes in log-intensity. Each pixel has two associated addresses, one for ON-spikes, one for OFF-spikes. An ON-spike means that the intensity has increased by a fixed relative amount, an OFF-spikes means that the intensity has decreased by a fixed relative amount.

An abstracted schematic of the pixel is shown in Fig. 1.6. It uses an active logarithmic front-end, a self-clocked switched capacitor feed-back amplifier and two very simple two-transistor comparators. The transistor level pixel circuit is shown in Fig. 1.7.

The chip includes a programmable bias generator [77] that enables easy parameter adjustment, guarantees reliable output in changing working conditions and little chip-to-chip variability.

Compared to traditional cameras, this sensor has some useful and complementary properties.

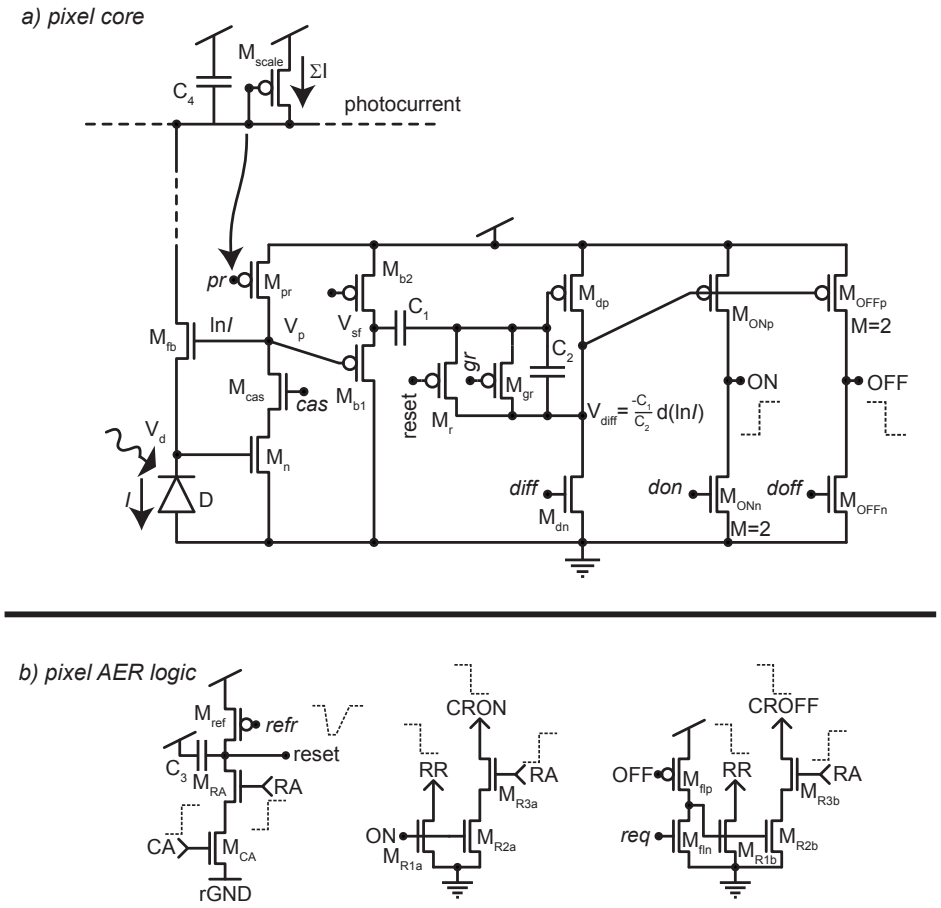


Figure 1.7: DVS128 transistor level pixel circuit, reprinted with permission from [39].

- Because it detects changes in log intensity, it reacts similarly to scene reflectance changes independent of illumination.
- Each pixel chooses its own exposure, resulting in a very high dynamic range of around 120dB.
- The pixel respond asynchronously to local scene reflectance changes, which results in very low latency, as low as $15\mu s$ in good lighting conditions.
- The sensor achieves low data rate due to complete temporal redundancy suppression right at pixel level.
- The DVS has low threshold mismatch due to capacitive amplification.

The DVS128 has been used in a variety of applications from high-speed robotics to car counting, proving that asynchronous, data-driven sensors can be very useful due to their pixel-parallel preprocessing that reduces latency and the load on subsequent processing stages [19, 78, 79, 20]. Research on the DVS is ongoing, both on the sensor side to increase sensitivity [80] and resolution, as well as on the software side to develop algorithms for processing the event-based data [81].

1.6 Neuromorphic Color vision

Neuromorphic color vision is mostly unexplored, all neuromorphic vision sensors presented in section 1.4 are monochromatic. However biology suggests that color information is an important cue, as most species have at least dichromatic color vision [82].

Color information is useful for example to distinguish shadows from real objects or to detect particular classes of objects based on color, e.g. stop signs.

Shimonomura and Yagi presented a color constancy system built from beam splitters and three silicon retinas [62]. This work demonstrates that logarithmic pixels are useful in color sensors because the output is more robust to changes of the light color. However, at this stage their color system is very bulky and not suitable for any robotic application.

In 2008, Olsson and Häfliger presented a spiking pixel based on a buried double junction (BDJ, see section 2.4) and therefore color sensitive [83]. This pixel is based on the monochromatic intensity-to-frequency pixel presented by Culurciellco et al. [54]. The pixel presented by Olsson and Häfliger has two output spike streams, one for each junction of the BDJ. The spiking frequencies code the light intensity while the ratio of the spiking frequencies encodes the light color. This pixel has the inherent disadvantage that for extraction of the light color information, post-processing is necessary.

In 2009, Fu and Titus presented a continuous time pixel also based on a BDJ [84]. This pixel has two output pathways, one for intensity using Delbruck's adaptive photo-receptor [85], the other output pathway encodes the light color in a voltage.

The pixel shows reasonable sensitivity, but the circuit relies on current copying, which is inherently mismatch-prone. This design is therefore not suited to use in an array.

1.7 Achievements presented in this thesis

The goal of this thesis was to develop the first event-based color silicon retina using standard CMOS technology. How to achieve color sensitivity in standard CMOS is described in chapter 2. In chapter 3, a first neuromorphic color vision sensor with built-in pattern-detection circuits is described [86]. However, to increase the flexibility we decided to focus on sensing color information without on-chip pattern-detection capabilities. Chapter 4 describes four different novel pixel circuits which respond asynchronously to color changes. The most promising pixel architecture has been analyzed in detail (chapter 5) and results have been published [87]. The pixel has been evolved and combined with a dynamic intensity change pathway (chapter 6, [88]). The final pixel architecture is presented in chapter 8. It provides three different pathways:

- The first pathway responds asynchronously to log intensity changes, equal to the DVS pixel [66].
- The second pathway responds asynchronously to color changes. The color sensitive pathway emits REDDER events if the mean wavelength of the incident light has increased or BLUER events if the wavelength has decreased.
- The third pathway finally reports log intensity values in a sustained way.

Chapter 8 shows measurement results that demonstrate the functionality of all three pathways, however performance does not reach a level that allows use in practical applications. Chapter 9 proposes circuit modifications to improve performance.

In conventional computer vision, color information is usually used to classify and distinguish objects. The utility of color change events is still unexplored, but it could be very useful for example in robotics to distinguish shadows (no color change, but intensity change) from real objects (color change). It might also help in situations where the DVS does not see borders between two objects because their reflectance is very similar even though their color is different. Modeling is needed to understand how exactly the color change events could be used. However, this situation is analogous to that of the DVS128, which was developed before application scenarios were conceived.

Chapter 7 presents the second main achievement of this thesis, a new word-serial AER scheme that speeds up event communication compared to the DVS128. The DVS128 suffers from relatively low per-pixel communication bandwidth for busy scenes. The new communication circuits allow us to scale up future event-based vision sensors to higher resolution without sacrificing per-pixel bandwidth. Additionally, in the new scheme a malfunctioning pixel can not block the communication

by constantly claiming the bus. Instead the malfunctioning pixel will emit events with a rate defined by a refractory bias parameter, which will usually result in a few thousand events per second.

Appendix A presents the PCB developed to mount and test the chips presented in chapters 6 and 8. Appendix B describes the measurement setup used to characterize the chips. Appendix C presents a short tutorial on LogSpice, a set of Matlab methods developed by Boahen's group at Standford to simulate asynchronous circuits. LogSpice has been used to simulate the new AER circuits presented in chapter 7.

1.8 Weak inversion

Most of the circuits presented in this thesis rely on MOS transistors running in the sub-threshold regime. Throughout the text, we approximate sub-threshold transistor currents by the equations presented by Vittoz and Fellrath [89, 90].

The saturation currents of nFET and pFET transistors are

$$I_{\text{nfet}} = I_0 e^{\frac{\kappa V_g - V_s}{U_T}} \tag{1.1}$$

$$I_{\text{pfet}} = I_0 e^{\frac{\kappa (V_b - V_g) - (V_b - V_s)}{U_T}}, \tag{1.2}$$

where U_T is the thermal voltage kT/q, I_0 is the off-current, which is directly related to the threshold; and κ is the sub-threshold slope factor, which is assumed to be constant, but in reality changes slightly with the current level. V_g, V_s and V_b represent the voltages of the gate, the source and the bulk relative to the substrate.

Chapter 2

Color sensitive light sensing

This chapter discusses how color sensitivity can be achieved cheaply with standard CMOS processes. It starts with a review of the device physics of photodiodes (other types of photodetectors like phototransistors or CCDs are not described in this thesis; in case of interest please refer to [91, 92]). Section 2.2 is dedicated to how spectral sensitivity is achieved traditionally. The last two sections introduce an alternative way of achieving spectral sensitivity by exploiting basic properties of silicon.

This chapter requires basic understanding of semiconductor device physics, primarily p-n junctions [92, 93, 94].

2.1 Photodiodes

If light is shining onto a piece of semiconductor, the energy of a photon can be absorbed by an electron, exciting it from the valence band to the conduction band, which corresponds to the generation of an electron-hole pair. The energy of the photon has to be larger than the band-gap of the semiconductor, which limits the wavelengths a given semiconductor is sensitive to. The generation rate of mobile carriers is given by

$$G(x) = \Phi_0 \alpha e^{-\alpha x}, \tag{2.1}$$

where Φ_0 is the photon flux penetrating the semiconductor surface, x is the distance from the surface and α is the absorption coefficient, which is strongly dependent on wavelength. Here we will concentrate on silicon as semiconductor. Fig. 2.1 shows the absorption coefficient for silicon. Fig. 2.2 shows the generation rate versus distance from the silicon surface for three different wavelengths of visible light.

The photon flux is calculated from the photon wavelength λ and the incident optical power P_{opt}:

$$\Phi_0 = (1 - R_\lambda)\frac{\lambda}{hc}\frac{P_{\text{opt}}}{A}, \tag{2.2}$$

where R_λ is the reflection coefficient of the semiconductor surface for a given wavelength, A is the area of the photodiode, h the Planck constant, and c the speed of

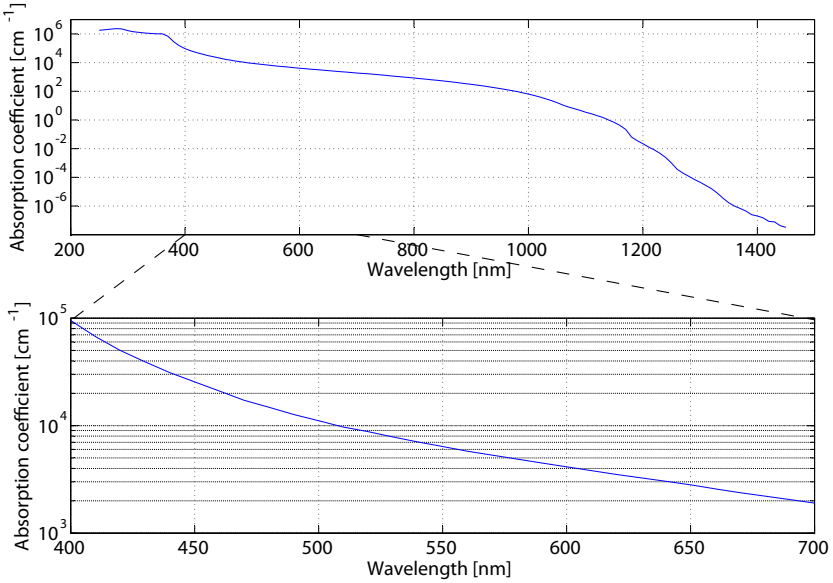


Figure 2.1: Absorption coefficient of silicon, data from [95]. The top plot shows the absorption coefficient over a broad range of wavelengths, the lower plot shows the absorption coefficient in the range of visible light.

light.

If light is shining onto a p-n junction which is biased at constant voltage, a photocurrent superimposed on the normal current is generated. This shifts the I-V curve as illustrated in Fig. 2.3. In the fourth quadrant, the photodiode works in photovoltaic mode and generates power. If photodiodes are used as photosensors, they are usually reverse-biased to be in the third quadrant.

There are several ways photodiodes are used in vision sensor circuits:

- The voltage over the photodiode is reset to a large reverse bias, then the photodiode is open-circuited and the photocurrent discharges the parasitic photodiode capacitance. After a fixed time, the voltage over the photodiode is read out, usually using a source-follower buffer. This is the most commonly used configuration; it is called the active pixel sensor (APS) [43]. The basic APS circuit, illustrated in Fig. 2.4, consists of a photodiode and three transistors per pixel.

- The second configuration is very similar to the APS pixel, but instead of integrating over a fixed time, the current is integrated over a fixed voltage range. The integration time then contains the information of the illumination [54, 56, 57, 58, 59, 67].

- In the third quadrant, the photodiode can be regarded as a light dependent

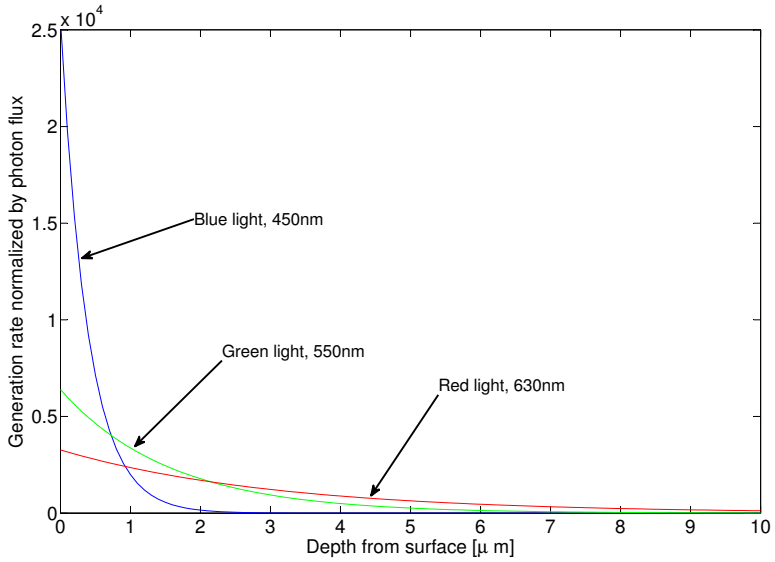


Figure 2.2: Generation rates of electron-hole pairs dependent on distance to silicon surface for different wavelengths. Usual junction depths of p-n junctions range from around $0.2\mu m$ for drain/source diffusions to a few micrometers for n-well to substrate junctions.

current source, because the current is only weakly dependent on the reverse bias. We use the photodiode mostly in this regime, therefore we concentrate our analysis on it.

2.1.1 Photocurrent calculation

For the following analysis, basic understanding of p-n junctions is assumed. Here we analyze the effect of light shining on a reverse biased p-n junction.

For the following calculations, we assume an abrupt step junction with constant doping concentrations in each the p- and the n-doped region. Additionally, we assume that the photodiode is a n-well to substrate junction like used in the DVS, which means that the n-type region is at the surface and much higher doped than the p-type substrate.

The generation rate of electron-hole pairs in the diode is given by equation 2.1, but how these carriers contribute to the overall photocurrent depends on the region the carriers are generated in. If the carriers are generated in the depletion region, the electron-hole pair is separated by the electric field and the carriers are swept across the junction (electrons to the n-type region, holes to the p-type region), so almost all carriers generated in the depletion region contribute to the photocurrent. If the carriers are generated in the neutral region, they will only contribute to the photocurrent if the minority carrier diffuses to the depletion region edge, where

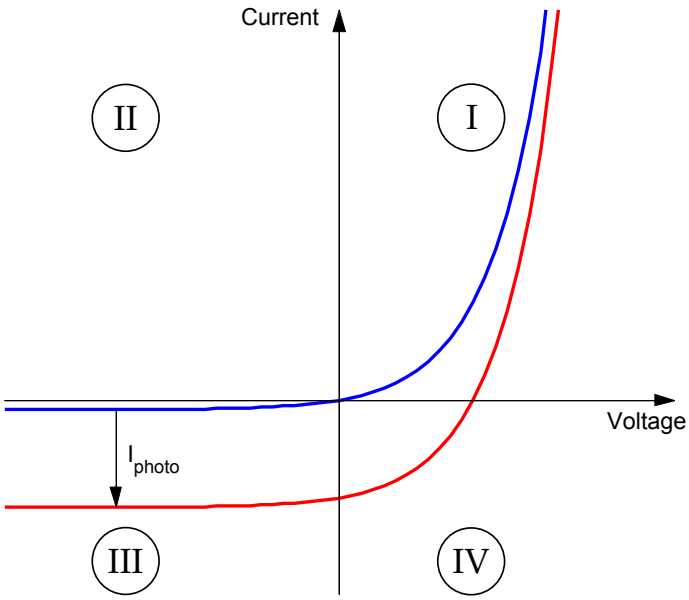


Figure 2.3: Photodiode I-V curve, arbitrary units.

it is swept across the junction. If it does not reach the depletion region edge, it will finally recombine with a majority carrier. So not all carriers generated in the neutral region will contribute to the photocurrent.

As described in the last paragraph, the photocurrent consists of a drift current of carriers generated in the depletion region and two diffusion currents of carriers generated in the neutral regions. The drift current density in the junction is simply the integral of the carrier generation over the depletion region W multiplied with the elementary charge q:

$$J_{\text{drift}} = q \int_W G(x)\,\mathrm{d}x = q \int_W \Phi_0 \alpha e^{-\alpha x}\,\mathrm{d}x. \tag{2.3}$$

Because the n-type region is much higher doped than the p-type substrate, the depletion region width on the n-type doped side is approximately zero. Thus we only have to consider the depletion region in the p-substrate:

$$J_{\text{drift}} = q \int_{x_j}^{x_j+w_d} \Phi_0 \alpha e^{-\alpha x}\,\mathrm{d}x, \tag{2.4}$$

where x_j is the depth of the junction edge and w_d is the depletion width in the p-substrate, as illustrated in Fig. 2.5. Solving this integral leads to

$$J_{\text{drift}} = q\Phi_0 \left(e^{-\alpha x_j} - e^{-\alpha(x_j+w_d)}\right). \tag{2.5}$$

The depletion region width can be calculated by

$$w_d = \sqrt{\frac{2\epsilon_s}{qN_A}(V_{bi} - V)}, \tag{2.6}$$

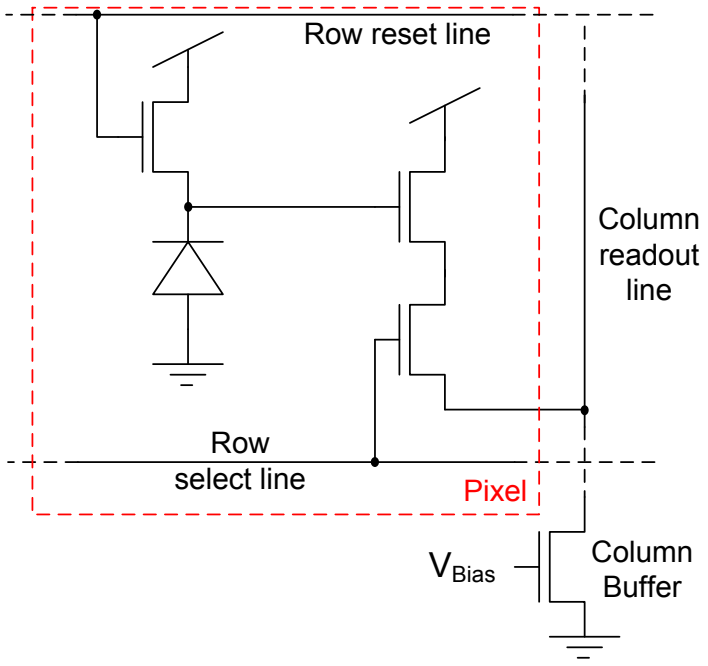


Figure 2.4: Active pixel sensor (APS) circuit

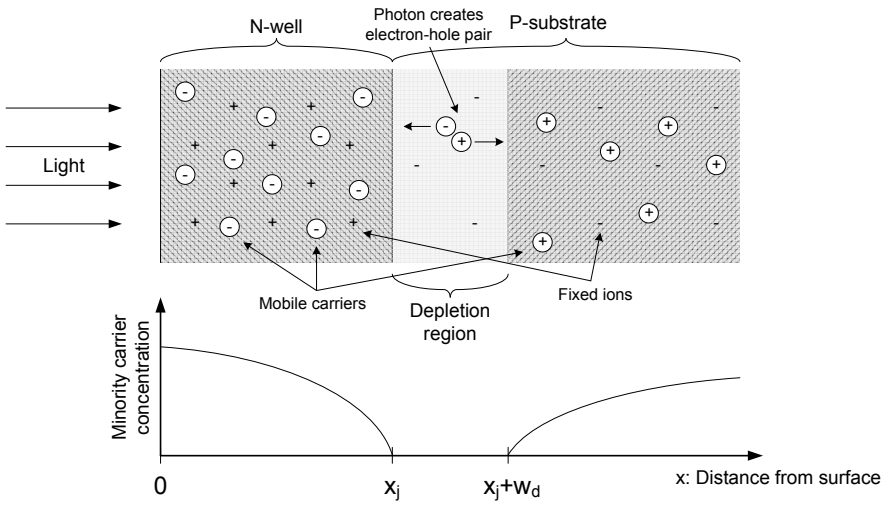


Figure 2.5: Photodiode illustration

where ϵ_S is the silicon permittivity, q the elementary charge, N_A the acceptor density in the substrate, V the applied voltage and V_{bi} the built in potential which is given by

$$V_{bi} = U_T log \frac{N_D N_A}{n_i^2}, \tag{2.7}$$

with N_D being the donor density in the well and n_i the intrinsic carrier density.

Because of the non-uniform generation of carriers in the neutral regions, a non-uniform carrier concentration will arise. Carries will then diffuse along the gradient of the carrier concentration. The carriers that diffuse to the edge of the depletion region contribute to the photo current.

To calculate the minority carrier concentration, we have to solve the diffusion equation in both neutral regions

$$-D_n \frac{\mathrm{d}^2}{\mathrm{d}x^2} n(x) + \frac{n(x)}{\tau_n} = G(x), \tag{2.8}$$

$$-D_p \frac{\mathrm{d}^2}{\mathrm{d}x^2} p(x) + \frac{p(x)}{\tau_p} = G(x), \tag{2.9}$$

where $n(x)$ and $p(x)$ are the respective minority carrier concentration as a function of the distance to the surface, D_n and D_p are the diffusion constants and τ_n and τ_p are the lifetimes of the minority carriers. The minority carrier lifetime is dependent on doping [96, 97]:

$$\tau = \left(\frac{1 + N/N_{\mathrm{ref}}}{\tau_0} + C_A N^2 \right)^{-1}, \tag{2.10}$$

where N is the doping concentration and τ_0, N_{ref} and C_A are fitting constants which are different for N or P type doped silicon.

To calculate the minority carrier concentrations, we need two boundary conditions for each region. For the n-well, the first boundary condition is that the carrier concentration has to be zero at the boundary of the depletion region, because the depletion region is completely depleted of mobile carriers. The second boundary condition is given by the rate of recombination on the surface, which is characterized by the surface velocity s. For the p-substrate, the first boundary condition is also that the carrier concentration has to be zero at the edge of the depletion region. But an additional boundary condition is necessary. We assume that the concentration goes to the thermal equilibrium at some distance x_e from the surface. If the substrate is thicker than a few diffusion lengths, this parameter is not very influential, because carriers generated far from the depletion region will most likely recombine before reaching the junction.

$$D_p \frac{\mathrm{d}}{\mathrm{d}x} p(x = 0) = s \cdot p(x = 0) \tag{2.11}$$

$$p(x = x_j) = 0 \tag{2.12}$$

$$n(x = x_j + w_d) = 0 \tag{2.13}$$

$$n(x = x_e) = \frac{n_i^2}{N_A} \tag{2.14}$$

Solving the diffusion equation with the boundary conditions given above for the n-well region leads to a hole concentration given by

$$p(x) = \frac{\tau_p \alpha \Phi_0}{2(L_p\alpha - 1)(L_p\alpha + 1)} \Bigg(- 2e^{-\alpha x} + \frac{e^{-x/L_p}\left(e^{-\alpha x_j} + \frac{L_p s}{D_p} e^{x_j/L_p} - \frac{L_p s}{D_p} e^{-\alpha x_j} + L_p \alpha e^{x_j/L_p}\right)}{\cosh \frac{x_j}{L_p} + \frac{L_p s}{D_p} \sinh \frac{x_j}{L_p}} + \frac{e^{x/L_p}\left(e^{-\alpha x_j} - \frac{L_p s}{D_p} e^{-x_j/L_p} + \frac{L_p s}{D_p} e^{-\alpha x_j} - L_p \alpha e^{-x_j/L_p}\right)}{\cosh \frac{x_j}{L_p} + \frac{L_p s}{D_p} \sinh \frac{x_j}{L_p}} \Bigg), \quad (2.15)$$

where L_p is the diffusion length of the holes in the n-Well, which is given by

$$L_p = \sqrt{D_p \tau_p}. \quad (2.16)$$

The minority carriers that reach the junction edge are swept across the depletion region and contribute to the total current. The diffusion current density is thus proportional to the derivative of the minority carrier concentration at the junction edge.

$$J_{\text{diffp}} = qD_p \frac{\mathrm{d}}{\mathrm{d}x} p(x = x_j) \quad (2.17)$$

$$J_{\text{diffn}} = qD_n \frac{\mathrm{d}}{\mathrm{d}x} n(x = x_j + w_d) \quad (2.18)$$

The total photocurrent is then

$$I_{\text{photo}} = A(J_{\text{drift}} + J_{\text{diffn}} + J_{\text{diffp}}), \quad (2.19)$$

where A is the area of the photodiode. From the above equations, it is possible to derive a closed formula for the photocurrent generated in the p-n junction. However, the formula is quite large and complicated and depends on process parameters which are generally not accessible to us. It is therefore not very helpful to state it here.

2.2 Color filters

The standard way of building color image sensors is to use color filter arrays on top of an array of monochrome pixels, e.g. each pixel receives and senses light of only one of three or four colors (usually red, green and blue). Fig. 2.6 illustrates the principle and shows the most commonly used pattern of color filters, called Bayer pattern. The missing colors at each photo-site are then interpolated. The color filters are often made out of polymers, which are deposited in additional process steps. These additional process steps (and other additional process steps to make more sensitive photodiodes) make special imager processes more expensive and also harder to get access to. Table 2.1 compares CMOS chip fabrication cost

for multi-project wafers (MPW) at the Europractice service for the UMC image sensor process and the corresponding mixed mode process. To our knowledge, the UMC CIS is world wide the only image sensor process with color filters available for MPW.

Table 2.1 clearly demonstrates that development cost is much higher when using the special image sensor process, because it is not available in the miniASIC program. Additionally to the higher price, the CIS also runs fewer times per year (twice versus five times).

Of course the image sensor process does not only offer color filters, it offers also other structures like micro-lenses and special implants to form more sensitive photodiodes with less dark current. However, at least in the UMC CIS there are restrictions on these special features which make them unusable for our circuits, for example minimal distance from special diodes to p-FETs.

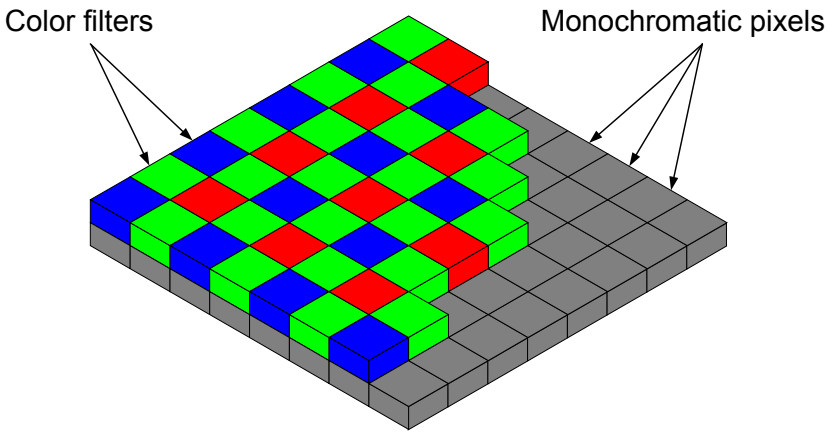


Figure 2.6: Illustration of a color filter array, adapted from Wikipedia [98].

Besides cost and limited availability, CMOS color imagers that use spatially-patterned color filters have three main disadvantages.

1. Up to two thirds of the light is lost at each photo-site [100], which reduces low light capabilities.

2. Color information is spatially aliased. Usually anti-aliasing filters are deposited on top of the imager, which reduce spatial resolution. Otherwise, extensive post-processing is needed to remove the aliasing artifacts.

3. The interpolation of color information requires the application of a computationally expensive Bayer decomposition algorithm, which limits low power application possibilities.

	UMC 180nm MM Mixed Mode		UMC 180nm CIS Imager
	standard MPW	miniASIC	
Price per mm^2 (Euros)	592	1092	816
Minimal size (mm^2)	5×5	1.525×1.525	5×5
Minimal cost (Euros)	14 800	2540	20 400

Table 2.1: Minimal CMOS process cost for MPW services through the Europractice service. Prices taken from Europractice [99] in May 2011. The CIS is not available in the miniASIC program.

2.3 Vertical color separation

Because the absorption coefficient of photons in silicon is wavelength dependent [92, 95] (Fig. 2.1), photons with shorter wavelengths are absorbed closer to the surface than photons with a longer wavelength. This means that measuring photocurrents at different depths in silicon gives information about the spectral composition of the incident light.

As an alternative to color filters, circuits that use the wavelength-dependent absorption length in silicon to measure color have been proposed in the 1980's [101, 102, 103], and many different circuits which employ this principle have been published [83, 84, 104, 105, 86, 106].

To our knowledge, only one company has used this principle in a commercial device. Foveon Inc. builds color image sensors [107, 108, 109] with three junctions per pixel at different depths. Color image sensors require three junctions at different depths to acquire RGB (red-green-blue) color images. Junctions at three different depths are not available in standard CMOS, but are available in triple well processes. However, to achieve good image quality, special process technology is required [109], since the color separation of normal triple well processes is not sufficient.

2.4 The Buried Double Junction

The buried double junction (BDJ) is the simplest structure which allows wavelength-sensitive light sensing based on the absorption length. It consists of a stack of two photodiodes formed by the active-well and well-substrate junctions, which is available in any CMOS process (Fig. 2.7). Because the absorption coefficient of photons in silicon changes monotonically with wavelength (Fig. 2.1), the ratio of the photocurrents generated by the two junctions is monotonically increasing, which is shown in Fig. 2.8 for the 0.5μm 3M 2P process used for the ColTmpDiff chip

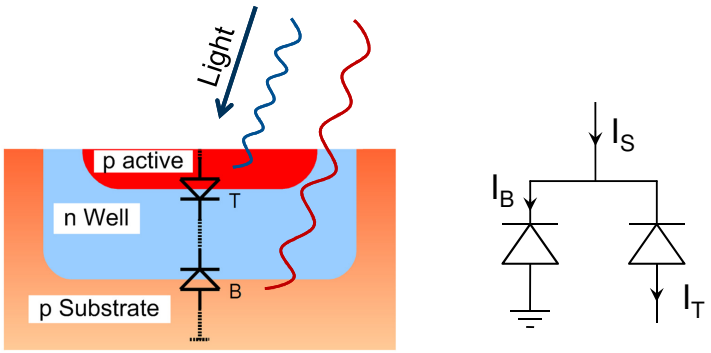


Figure 2.7: Illustration and schematic of the buried double junction. Figure adapted from [106].

described in chapter 4.

The buried double junction allows only dichromatic light sensing, because it offers only two photodiodes. For standard color imaging, three photodiodes would be needed, but a vertical stack of three photodiodes is not available in standard CMOS. We believe some basic dichromatic wavelength separation capability is sufficient for many vision tasks, which is supported by the fact that most animals are only dichromats [82]. Therefore this thesis investigates circuits that use a BDJ.

2.4.1 Estimating spectral sensitivity of a BDJ

The spectral sensitivity of a BDJ is the change in the ratio of the junction currents for a given change of light wavelength:

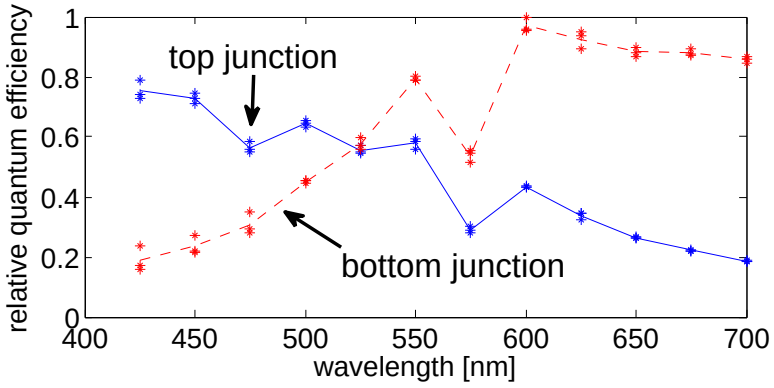
$$s_c(\lambda) = \frac{d}{d\lambda}\frac{I_S}{I_T}. \tag{2.20}$$

If step junctions are assumed (this is of course an approximation, but diffused junctions can only be solved numerically [94]), it is actually possible to calculate closed formulas for both junction currents, similar to section 2.1 and actually done by Sedjil et al. [111]. However, the resulting formulas are so long and complex that they are difficult to interpret.

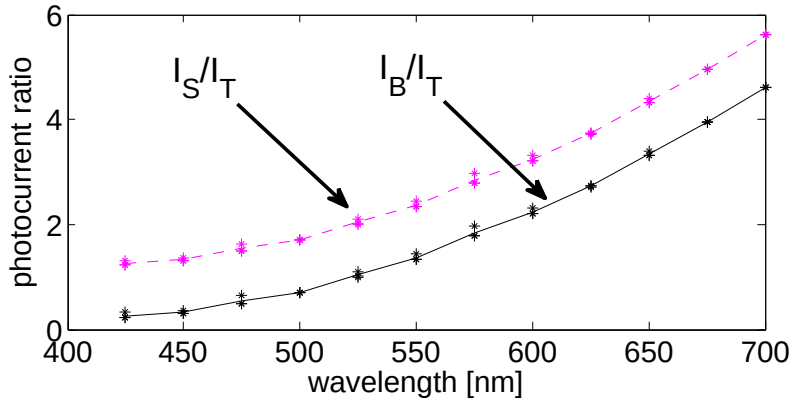
A very simple model of the color sensitivity can be derived based on the assumption that the upper junction collects all the charge generated from the surface to the middle of the quasi-neutral region of the n-well, while the lower junction collects the charge generated from the middle of the quasi-neutral region of the n-well to infinity. x_t is the junction depth of the upper junction, x_b is the junction depth of the lower junction and W_t is the depletion region width of the upper junction. These values are illustrated in Fig. 2.9.

The junction current densities are then:

$$J_T = q\int_0^{(x_t+W_t+x_b)/2} G(x)\,\mathrm{d}x = q\int_0^{(x_t+W_t+x_b)/2} G(x)\Phi_0\alpha e^{-\alpha x}\,\mathrm{d}x \tag{2.21}$$



(a) The relative quantum efficiency (normalized to the maximum value) of the top and bottom junction. The wiggles in the curves probably stem from reflections in the oxide [110].



(b) Ratio of the photocurrents; the solid line shows the ratio between the junction currents; the dashed line shows the ratio of the available currents I_S/I_T.

Figure 2.8: Spectral response of the BDJ in the $0.5\mu m$ process used for the chip presented in chapter 4.

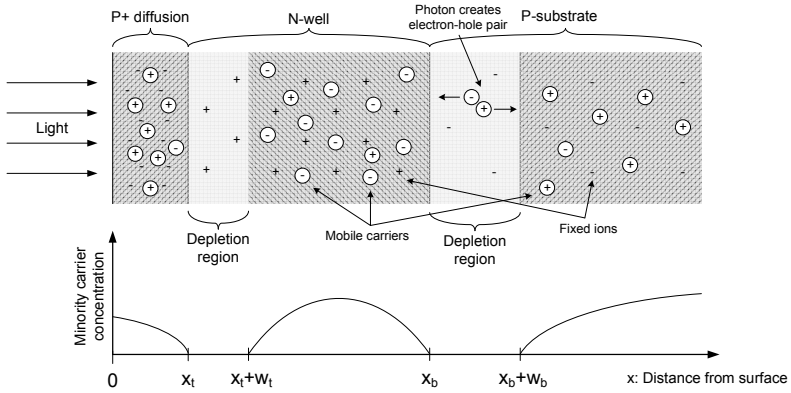


Figure 2.9: BDJ illustration showing minority carrier concentration.

$$J_T = q\Phi_0\left(1 - e^{-\alpha(x_t+W_t+x_b)/2}\right) \tag{2.22}$$

$$J_B = q\int_{(x_t+W_t+x_b)/2}^{\infty} G(x)\,\mathrm{d}x = q\int_{(x_t+W_t+x_b)/2}^{\infty} G(x)\Phi_0\alpha e^{-\alpha x}\,\mathrm{d}x \tag{2.23}$$

$$J_B = q\Phi_0\left(e^{-\alpha(x_t+W_t+x_b)/2}\right) \tag{2.24}$$

The ratio of the available currents is thus

$$\frac{J_B + J_T}{J_B} = \frac{1}{1 - e^{-\alpha(x_t+W_t+x_b)/2}}. \tag{2.25}$$

The color sensitivity then follows:

$$s_c(\lambda) = \frac{d}{d\lambda}\frac{J_B + J_T}{I_T} \tag{2.26}$$

$$= \frac{d}{d\alpha}\frac{J_B + J_T}{I_T}\frac{d\alpha}{d\lambda} \tag{2.27}$$

$$s_c(\lambda) = \frac{-\frac{x_t+W_t+x_b}{2}e^{-\alpha(x_t+W_t+x_b)/2}}{\left(1 - e^{-\alpha(x_t+W_t+x_b)/2}\right)^2}\frac{d\alpha}{d\lambda}. \tag{2.28}$$

We compared our simple model to simulations of a step junction model (derived from similar equations as in subsection 2.1.1). We simulated in Matlab resulting carrier concentrations, quantum efficiencies and photocurrent ratios on the assumption of step junctions for different parameters. We chose the junctions depths and doping concentrations based on the process specifications for the $0.5\mu m$ process used in chapter 4 and the $180nm$ process used for the chips presented in chapters 6 and 8. Table 2.2 lists the parameters and the resulting photo-current ratio change.

Fig. 2.10 shows the results of the simulations and illustrates that decreasing the junction depth and increasing the doping concentrations should increase the color sensitivity. Intuitively, if the well is shallower, the ratio between top and bottom junction current changes in favor of the bottom current for longer wavelengths,

Parameter	Fig. 2.10(a) $0.5\mu m$ process	Fig. 2.10(b)	Fig. 2.10(c)	Fig. 2.10(d) $180nm$ process
Drain/source diffusion doping N_{p+}	$8 \cdot 10^{19} cm^{-3}$	$8 \cdot 10^{19} cm^{-3}$	$10^{21} cm^{-3}$	$10^{21} cm^{-3}$
n-Well doping N_{nw}	$6 \cdot 10^{16} cm^{-3}$	$6 \cdot 10^{16} cm^{-3}$	$4 \cdot 10^{17} cm^{-3}$	$4 \cdot 10^{17} cm^{-3}$
Substrate doping N_{psub}	$5 \cdot 10^{15} cm^{-3}$	$5 \cdot 10^{15} cm^{-3}$	$10^{15} cm^{-3}$	$10^{15} cm^{-3}$
Upper junction depth x_t	$0.3\mu m$	$0.2\mu m$	$0.3\mu m$	$0.2\mu m$
Lower junction depth x_b	$3\mu m$	$1.8\mu m$	$3\mu m$	$1.8\mu m$
Drain/source diffusion bias voltage V_a	$0.8V$	$0.8V$	$0.8V$	$0.8V$
n-Well bias bias voltage V_w	$1V$	$1V$	$1V$	$1V$
Surface velocity s	$10^4 cm/s$	$10^4 cm/s$	$10^4 cm/s$	$10^4 cm/s$
Incident light	$1mW/m^2$	$1mW/m^2$	$1mW/m^2$	$1mW/m^2$
Current ratio change from 430nm to 650nm	2.61	3.52	2.90	3.97

Table 2.2: Simulation parameters used for the BDJ photo-current simulations shown in Fig. 2.10. Fig. 2.10(b) uses doping concentrations from the $0.5\mu m$ process and junction depths from the $180nm$ process, Fig. 2.10(c) vice-versa.

but does not change very much for short wavelengths, because most of the short wavelength photons are absorbed close to the surface. This means that the change in the current ratio should increase with a shallower well.

The elaborate simulation and the simple model compare surprisingly well. Thus we expect that color sensitivity increases with decreasing well to substrate junction depth.

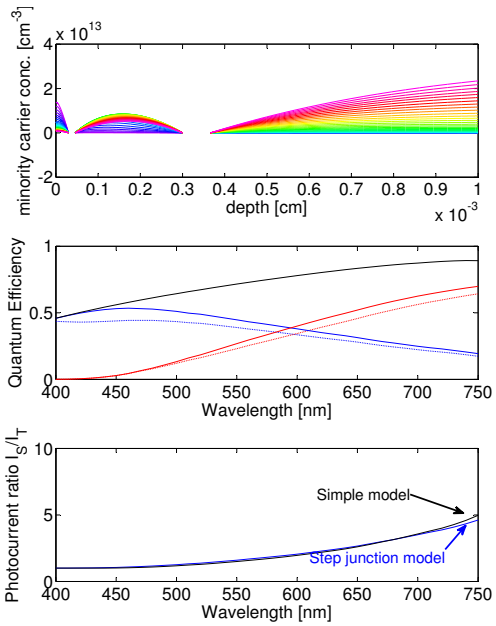
Contrary to these simulations, comparing the measured spectral sensitivity of the $0.5\mu m$ process to the $180nm$ process shows that the $0.5\mu m$ process is actually more sensitive (Fig. 2.11) despite the deeper junctions. Because we do not have access to the doping profiles, we can only speculate that it could be due to different surface velocity or minority carrier lifetimes not following the model found in the literature [96, 97], different retrograde doping or some effect of different epitaxial layer thickness. This result also shows us that the derivation of formulas based on the assumption of step junctions and with several unknown parameters is not very helpful to estimate real spectral sensitivities.

2.4.2 Measuring spectral sensitivity of the BDJ

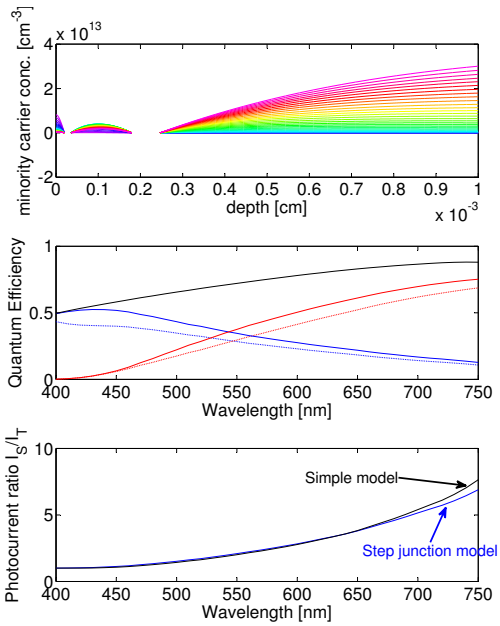
The circuit used to measure the spectral response of the $0.5\mu m$ and the $180nm$ process is based on a circuit presented by Fasnacht and Delbruck [106]. The circuit is shown in Fig. 2.12.

The circuit outputs a pulse-width modulated signal whose frequency codes the irradiance and whose duty cycle codes the spectral content of the light. This measurement circuit holds the BDJ photodiodes reverse-biased at constant voltages with the negative feedback configuration of M_{N1}, M_{N2} and M_{P3} (and M_{P1}, M_{P2} and M_{N3} respectively). The circuit is controlled by a simple state machine with two states. In the first state M_{Nsw} is conducting, which bypasses the capacitor C_T, while the photocurrent I_S is integrated on C_S until the threshold voltage V_{thresSum} is reached. Then the state machine switches to the second state. In the second state, M_{Psw} is conducting and thus the charge on C_S is reset. M_{Nsw} is not conducting, I_T is integrated on C_T until V_{thresTop} is reached. The photocurrents can then be inferred from the time taken to reach threshold and the capacitor value. Example traces from this circuit implemented in the 180nm process are shown in Fig. 2.13 for two different light colors. The difference in the duty cycle of the *State* signal is clearly visible.

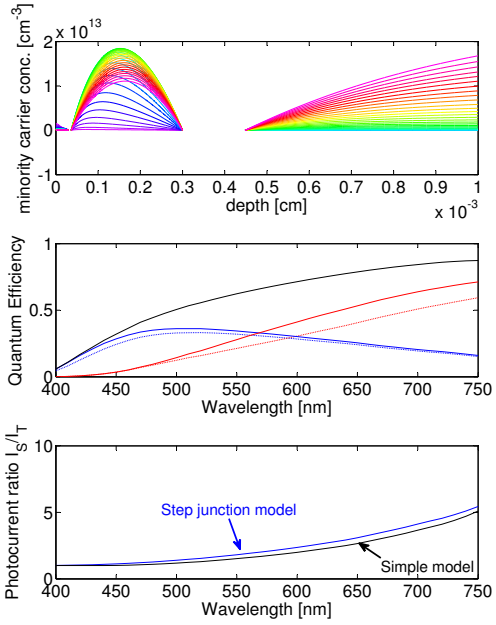
In our measurement circuit on the 180nm cDVSTest10 chip we used a pixel-sized BDJ rather than the very large BDJ used on the ColTmpDiff chip. Therefore the photocurrents are quite small and so the off-current leakage in the switch transistors M_{Nsw} and M_{Psw} is larger than the photocurrent except in bright light conditions. Because our monochromator does not produce sufficient light intensity, it was not possible to measure a continuous spectral sensitivity curve. However, by using a variety of brighter colored light sources, we could compare the BDJ spectral sensitivities of the 0.5μm process and the 180nm process (Fig. 2.11). This measurement shows that the 180nm process is slightly less sensitive to color changes than the



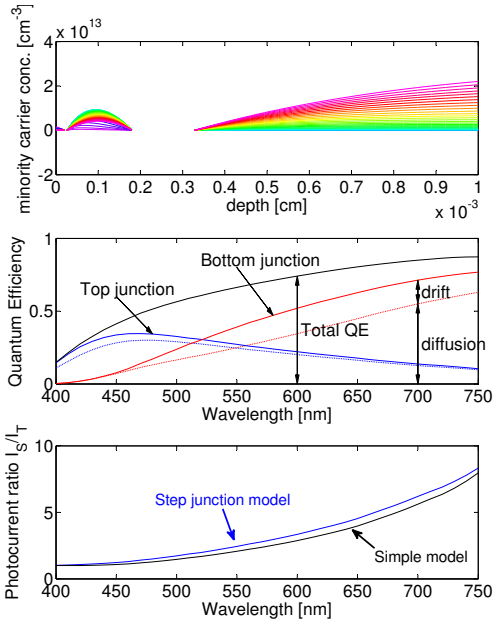
(a) Parameters from the $0.5\mu m$ process



(b) Doping concentrations from the $0.5\mu m$ and junction depths from the $180nm$ process

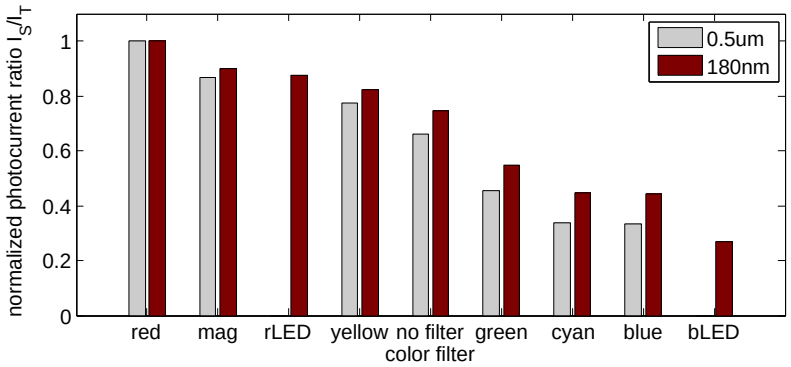


(c) Doping concentrations from the $180nm$ and junction depths from the $0.5\mu m$ process

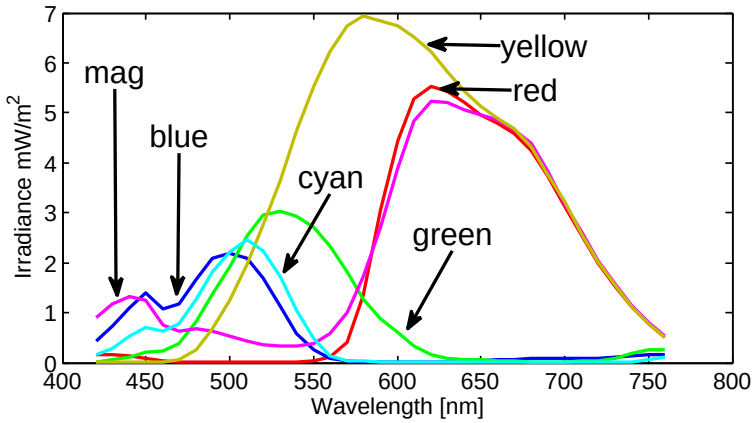


(d) Parameters from the $180nm$ process

Figure 2.10: Minority carrier concentration for different light wavelengths from 400 to 750 nm (the color is coded approximately to the color in the spectrum), quantum efficiency and current ratio in step junction BDJ for different doping concentrations and junction depths. The current ratio plot compares the more realistic model with the simple model from equation 2.25. The parameters for (a) are taken from the $0.5\mu m$ process specification, the ones for (d) from the $180nm$ process. Subplots (b) and (c) use junction depths from one process and doping from the other.



(a) I_S/I_T of the $0.5\mu m$ and the 180nm process for different colored light sources. The ratio of the currents was normalized for easier comparison. The chip was illuminated with the powerful red (rLED, 630nm) and blue (bLED, 450nm) LEDs used for pixel characterization and a strong white light source whose light color was changed using Edmund color filters (mag=magenta). In the 180nm process, I_S/I_T is 5.5 for red light and 1.5 for blue light.



(b) The light transmitted by the different color filters. The curves show the spectral response of the color filters multiplied with the spectral content of the light source.

Figure 2.11: Comparison of the spectral response of the BDJ in the 0.5μm and the 180nm process.

0.5μm process, but the I_S/I_T ratio in the 180nm process still varies by a factor of 3.7 over the visible range of wavelengths.

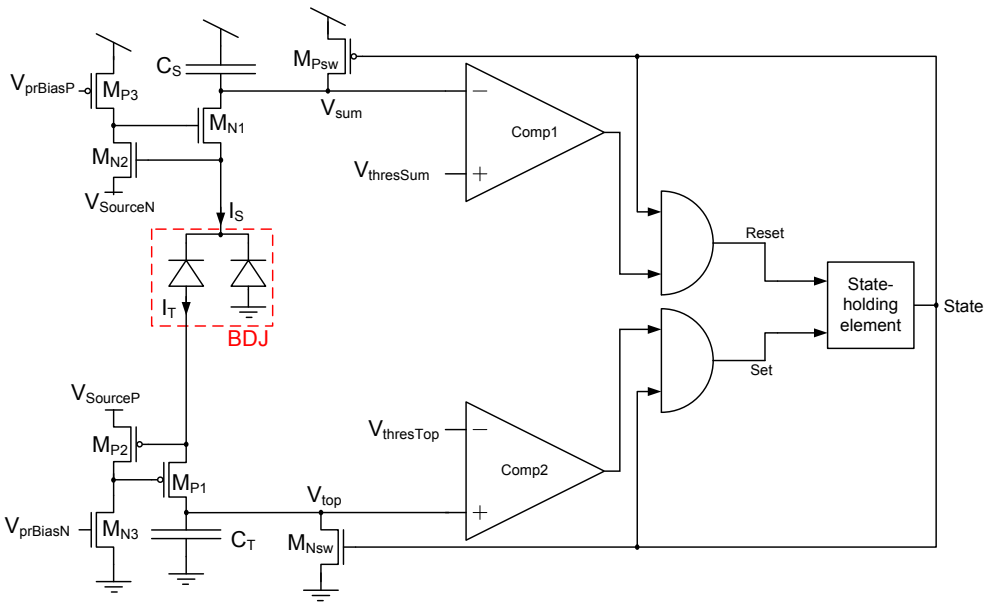
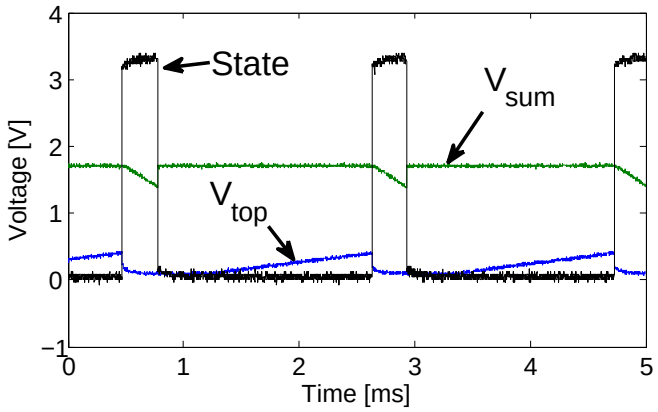
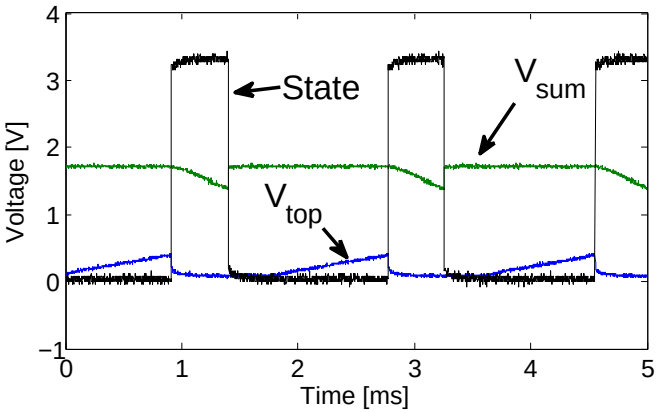


Figure 2.12: Circuit used for measuring the spectral sensitivity of a BDJ.



(a) Scope traces of V_{sum}, V_{top} and *State* for red light.



(b) Scope traces of V_{sum}, V_{top} and *State* for blue light.

Figure 2.13: Scope traces of V_{sum}, V_{top} and *State* of the spectral sensitivity circuit for blue and for red light. It is clearly visible that the duty cycle of the *State* signal is longer for blue light.

Chapter 3

DollBrain1: The first multi-pixel spiking color vision sensor

DollBrain1 has been developed in the course of the Swiss National Science Foundation funded project 'Micropower Face and Voice detection', aiming to build an integrated low power human interaction detector. The idea was to combine two relatively unreliable low power sensors, a vision sensor for face detection and an auditory sensor for speech detection, to achieve reliable human interaction detection. A speech detection chip has been built by a master student and was presented at the IEEE International Symposium of Circuits and Systems [112].

The DollBrain1 chip was our first (and only) attempt to build a low power face detector. It was the first chip developed and fabricated in the course of this thesis and also the first chip I developed and designed.

DollBrain1 is a dichromatic time-to-first-spike imager and a simple pattern detector test chip which uses BDJs to measure intensity and chromatic information. The pattern detection is neither size nor shift invariant but the idea was that due to the intended application scenario of hand-held devices or toys, size and position of the face is constrained. It should be possible to extend the presented architecture to achieve a small degree of shift invariance by parallel implementation of pattern detector units. The proposed pattern detector combines basic face features and could in theory be extended to a more realistic face detector, but at the cost of hugely increased complexity.

3.1 Motivation

Custom digital and mixed signal face detection circuits have been developed which allow for skin color detection [113] or face detection [114]. These systems are highly efficient, but because they partition their operation into image sensor followed by

analog or digital processor, they still require system level power consumption of at least $100mW$, making it impossible to run them continuously under battery power. It may be desirable to burn full power only when the presence of a human desiring interaction is detected.

The high power consumption of traditional vision systems is partly due to their repetitive processing of highly redundant input. Avoiding the readout and post processing of every pixel by doing necessary processing directly on chip at the pixel level can reduce the power consumption significantly. This approach could be feasible for basic pattern recognition especially if the detection of a pattern can wake up more power-hungry but more reliable post processing when it is likely to be needed.

The DollBrain1 chip uses an architecture that combines focal plane face pattern detection with dichromatic imaging capability that can be used for more sophisticated post processing after wake up. The face detection is based on the fact that skin has high reflectance in the near IR and that the intensity distribution coming from a face has prominent dips at the eyes.

3.2 Pattern Detector Architecture

Our face pattern detector detects a face blob that is redder than its surrounding and the presence of "eyes" that are dark pixels around the upper middle of the face.

Angelopoulou showed that skin strongly reflects light in the near IR independent of skin color (race) [115]. Therefore the dichromatic pixel proposed here, which has the ability to discriminate bluish and IR spectral characteristics, should serve as a decent indicator of skin color.

Viola and Jones showed that the eyes are the most prominent features of a face in a monochromatic image [116, 117]. More specifically, their face detection algorithm uses a cascade of simple rectangle filters which are selected and trained by an AdaBoost learning algorithm. The first two (and therefore most important) filters illustrate the fact that the eyes are shadowed and appear darker than the cheeks and the nose due to their position in the skull. This means that the average of the pixels representing the eyes is darker than the average of pixels representing the cheeks or the nose.

Fig. 3.1 shows a schematic overview of the face pattern detector for one possible face position. The detector assumes the presence of a face if eye (E) pixel 2 is darker than pixel 3 and 7, eye (E) pixel 4 is darker than pixel 3 and 9 and the average color of face pixels (R pixels) is redder than the surrounding (B pixels).

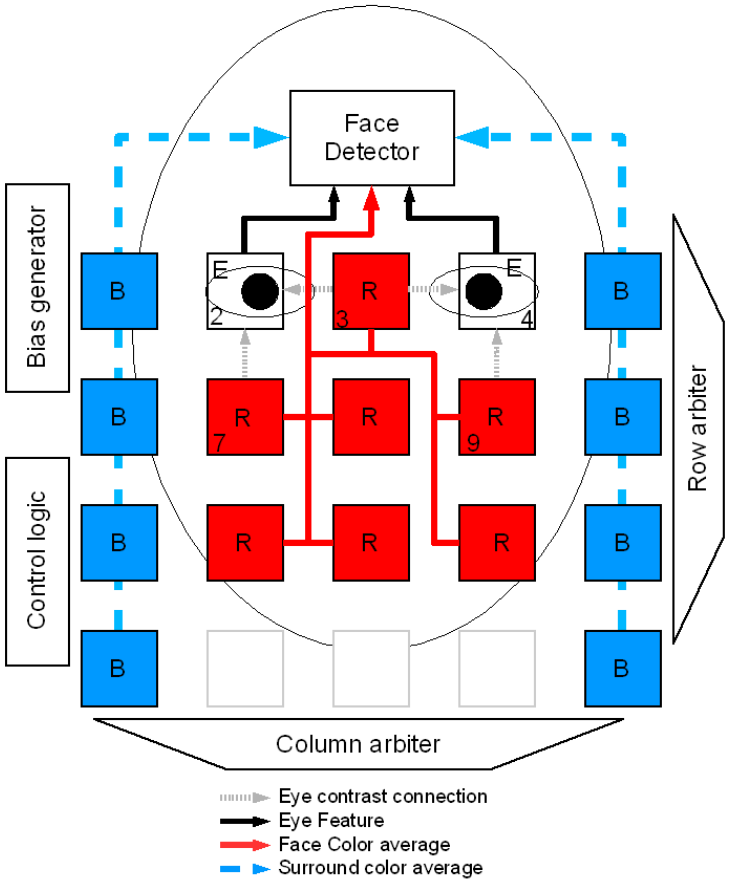


Figure 3.1: Architecture of the face detector.

3.3 Implementation

3.3.1 Pixel Architecture

The pixel sensor is based on the same structure as the dichromatic spectral measurement circuit proposed by Fasnacht and Delbruck [106] but utilizes a different sampling method to achieve a time encoding of the intensity and a voltage encoding of the ratio of the BDJ junction currents.

The pixel sensor structure consists of the buried double junction photodiodes L and U and two switches driven by control signal ϕ (Fig. 3.2). The intensity and chromatic information is sampled in two phases. In the first phase both photodiode capacitances are charged to reference voltages $V_A = V_{refL}$ and $V_C = V_{refH}$. In phase 2 the switches are opened and the photocurrents I_{phL} and I_{phU} discharge the parasitic photodiode capacitances C_L and C_U:

$$\Delta V_C = -\frac{I_{phL}}{C_L}\Delta t \tag{3.1}$$

$$\Delta V_A = \Delta V_C + \frac{I_{phU}}{C_U}\Delta t \tag{3.2}$$

Phase 2 ends by sampling V_A when V_C reaches an adjustable V_{Clo}. By measuring V_A and the time Δt to reach V_{Clo}, we can calculate $I_{phL} \cdot K_L$ and $I_{phU} \cdot K_U$, where K_i denotes a constant factor predominantly determined by the photodiode capacitance. However, for the face detection it is enough to determine if the face area has more spectral power in the red than the surrounding area and for that it is enough to measure V_A, because:

$$\Delta V_A = \Delta V_C \left(1 - \frac{I_{phU}}{I_{phL}}\frac{C_L}{C_U}\right), \quad \text{where} \quad \Delta V_C = V_{Clo} - V_{refH}. \tag{3.3}$$

Therefore the higher the voltage V_A at the sampling time, the bluer the scene, independent of intensity. As can be seen from equation 3.3, the color information signal is scaled with the capacitor ratio C_L/C_U. Therefore we added a capacitor from node V_C to ground, which slows down the integration but increases the color signal.

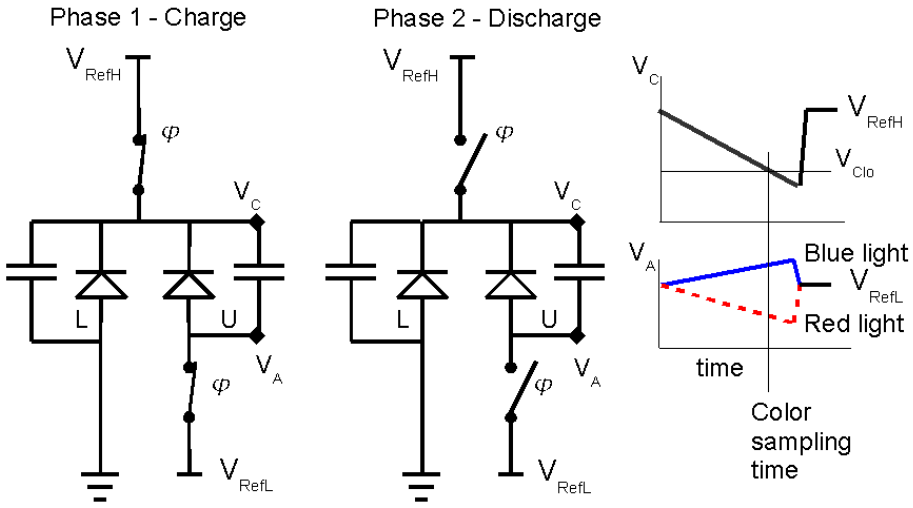


Figure 3.2: The two phases of the pixel operation. Phase 1 recharges nodes V_C and V_A to reference values. In phase 2 both photodiodes are discharged by the photocurrents. V_A depends on the ratio of the photocurrents and therefore on the spectral content of the light. Figure adapted from [106].

The pixel sensor is embedded in a communication and control structure (Fig. 3.3). The sampling phases are scheduled by a state machine. After all the pixels

complete phase 1 (determined by a global wired OR signal BUSY), phase 2 is started simultaneously in all pixels by the global *StartFrame* signal. Pixels then wait for the internal signal V_C to go below threshold V_{Clo} to conclude phase 2. Upon conclusion of phase two, the state machine signals via Address-Event-Representation (AER) communication to the periphery, and the pixel enters phase 1 again to prepare for the next integration cycle. The pixel circuit (Fig. 3.3) includes a sample and hold circuit for V_A and AER interfacing circuits.

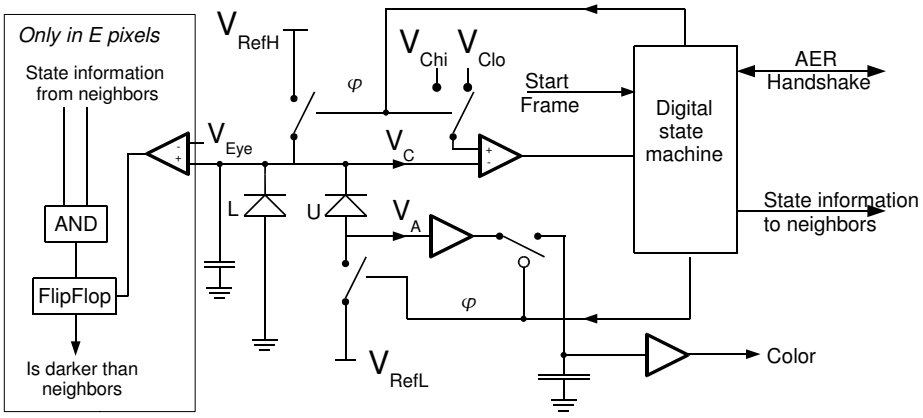


Figure 3.3: Pixel architecture, including control state machine, sample and hold for the color value and the circuits for the computation of the eye feature (section 3.3.2). The pixel consists of a buried double junction photodiode, two comparators, two source-followers, a capacitor and some digital control logic.

3.3.2 The Pattern Detector

The pattern detector unit is built by combining detection of three features, face/ surround color contrast, left eye and right eye luminance contrast. Color contrast is computed by capacitive averaging of color voltages – one averaging circuit each for surround and face color. The average face color is compared to the average surround color by a simple operational amplifier comparator and the comparator decision is latched on falling edge of BUSY.

When V_C of an E-pixel (Fig. 3.1) reaches an adjustable threshold $V_{eye} > V_{Clo}$ it samples the state (still integrating or finished integrating) of neighboring pixels. If the neighbors have already finished discharging, they are brighter and the eye is flagged as detected. V_{eye} sets the necessary contrast for the presence of an eye.

The three face features (color, left eye, right eye) are the input to an AND gate whose binary output indicates presence of the pattern.

3.3.3 Time-to-first-spike "imager"

The DollBrain1 sensor in conjunction with a speech detector was intended as a wake-up circuit for more powerful processing circuits. Thus we added some (of course due to the low pixel count very limited) imaging capability to this vision sensor. This capability is provided by extending the control state machine and adding AER interfacing circuits [76] to implement an enhanced time-to-first-spike encoding [58]. At the start of integration, a spike with a special address (frame start address) is emitted. As soon as a pixel reaches V_{Clo}, it emits a spike by putting its address on the bus. At this moment the color voltage is connected to the output as an analog signal and converted by an external analog-to-digital converter (ADC).

3.4 Test chip

A test chip with 20 pixels and one pattern detector unit was fabricated in a $1.5\mu m$ 2-metal 2-poly process. Fig. 3.4 shows a die micrograph, table 3.1 lists the specifications. The fill factor of the pixels is much lower than possible because the well and the active are completely surrounded by contacts, which is not necessary. By using more reasonable contacts, the fill factor could be increased easily to about 7.35% as illustrated in Fig. 3.5, removing the guard ring would increase the fill factor even more.

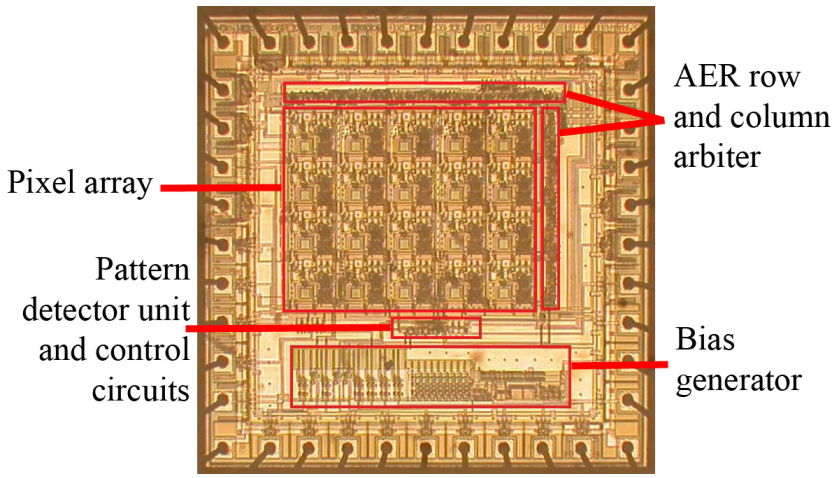


Figure 3.4: DollBrain1 die micrograph. Die size is 2.2 by 2.2 mm.

The chip includes a bias generator for fixed bias currents [118]. For testing and characterization, the chip is placed on a PCB with a Silicon Laboratories C8051F320 USB1.1 transceiver/micro-controller and an Analog Devices AD5391 16-channel DAC for generating reference voltages and the possibility to override internal bias voltages (Fig. 3.6). The system is interfaced to the jAER software [81].

Functionality	Pattern detector, asynchronous time-to-first-spike imager with analog dichromatic spectral value output
Pixel size μm (lambda) Fill factor	244x256 (305x320) 4.06 % (PD area 2540 μm^2)
Fabrication process	2M 2P 1.5um
Pixel complexity	99 transistors (11 analog), 3 capacitors
Array size	4x5
Interface	5-bit word-parallel AER
Dynamic Range	70dB
Power (@5V)	Analog: 30μA Digital @300 Hz frame rate: 14μA Pads: 550 μA (80% analog output pads for characterization)

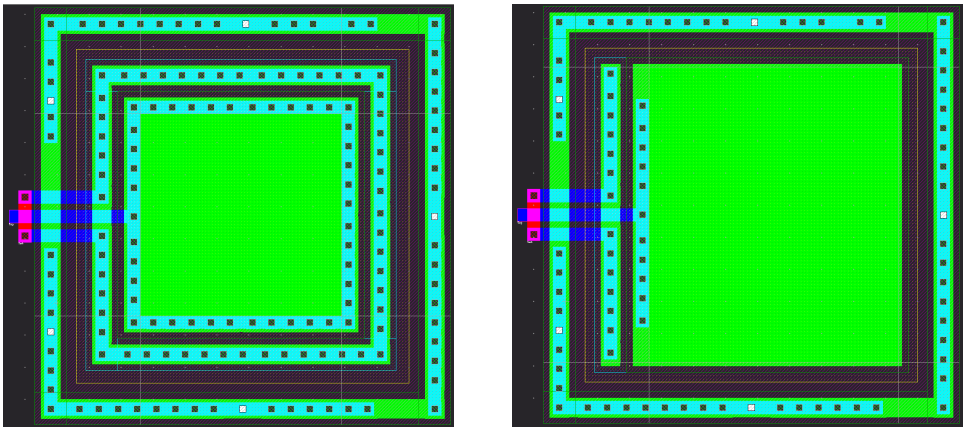
Table 3.1: DollBrain1 vision sensor specifications

3.5 Measurement Results

Fig. 3.7 shows the average integration time of all the pixels over more than 3 decades of irradiance. The measurements were conducted with an incandescent light source and Kodak Wratten neutral density filters. Eue to the IR component in the spectrum of the light source we measured and corrected the attenuation factor of the filters. The plot shows that the integration time is nearly inversely proportional to irradiance over more than 3 decades. At very high irradiance, integration time saturates due to finite circuit speed, at very low irradiance integration time saturates due to dark current. The plot also shows that the color value is varying about 5%. The shift towards red with decreasing intensity can be attributed to the fact that the neutral density filters are less effective in the near IR.

Fig. 3.8 displays the 8 bit encoding of monochromatic light for all 20 pixels. The encoding of the color saturates below 500nm and above 750nm. The blue limit could not be fully explored because we used an incandescent light source. However between 500nm and 750nm we get a good resolution encouraging further exploration.

Fig. 3.9 shows the integration time versus wavelength for all pixels normalized by irradiance. The data shows a strong fixed pattern which was observed in all dies. Fig. 3.10 shows measurements of the pattern detector circuits. The left plot shows the output of the capacitive averaging circuits when the sensor is aimed at a computer monitor. The computer monitor displays a stimulus similar to the one on



(a) DollBrain1 BDJ, 2540μm^2 light sensitive area.

(b) Improved layout for BDJ, 4592μm^2 light sensitive area.

Figure 3.5: Comparison of DollBrain1 BDJ layout and improved layout with more light sensitive area.

the right of Fig. 3.10, where either the hue of the "face" is held constant and the hue of the surround is stepped from 0 to 1 or vice versa. It can be seen that the average surround color voltage is higher for a given hue, which means that the detector is biased towards detecting a red blob even in uniform color stimulus, which is of course undesired.

The right plot of Fig. 3.10 shows the maximum V_{eye} threshold voltage, at which the sensor still can detect the eye feature for a given luminance ratio between E pixel and its neighbors. The measurements demonstrate that the pattern detector circuit works. Its performance can be improved by spending more area on the comparator and adding a threshold for the color feature.

3.6 Discussion

The chip provides the intended functionality of achieving dichromatic light sensing combined with very simple pattern detection.

However the performance is not satisfying. The pixel is slow due to the additional capacitor on V_C, the fixed pattern noise is high and the pattern detector is biased towards detecting the "redness" feature. The performance could be improved by relatively small changes to the circuit and more careful layout.

- Removing the capacitor on V_C increases the pixel speed at the expense of smaller signal swing of V_A.
- Increasing the fill factor by more reasonable contacting of the photodiode increases the photocurrent and therefore pixel speed and improves signal-to-noise ratio (SNR).

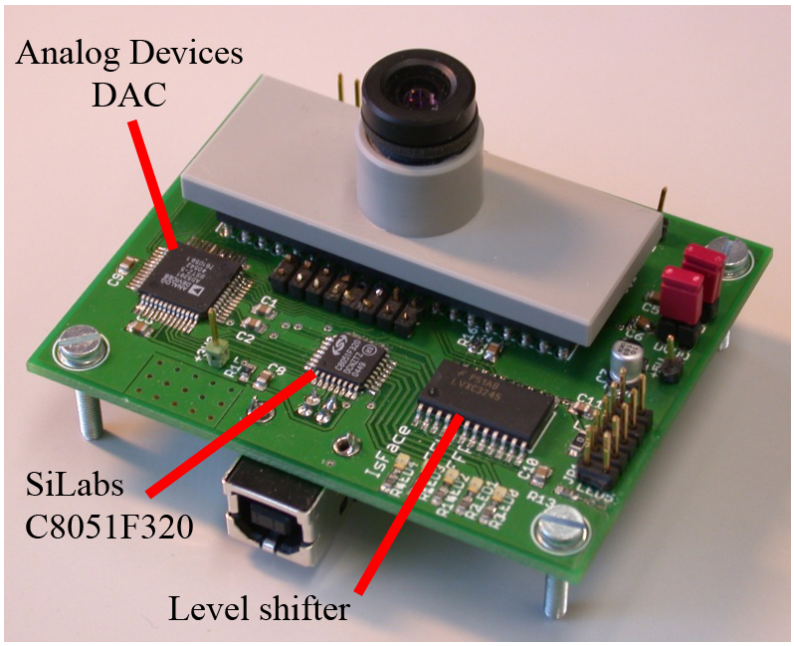


Figure 3.6: DollBrain1 mounted on PCB

- Adding a tunable threshold redness difference in the pattern detection allows to improve detection performance.
- The eye feature detection slows down the E pixels with the additional comparator gate, adding dummy transistors to the other pixels can remove some of the fixed pattern noise.
- More careful design and layout of the comparators and buffers and possibly dummy pixels at the border of the array will improve the fixed pattern noise.

However, the main disadvantage of this circuit is that this kind of pattern detection does not scale. It is possible to implement shift invariance by distributing many pattern detectors across a bigger array. This implementation of parallel pattern detectors would increase the wiring complexity a lot, but still the shift invariance achieved would only be on a rather coarse grid, faces which do not fall exactly on the detectors might go unnoticed. Adding some amount of size invariance or even more face features would increase the wiring complexity to a prohibitive level.

3.7 Conclusion

The circuit presented in this chapter shows encouraging color sensitivity, but the usefulness of such a simple pattern detector seems doubtful. Therefore the rest of this thesis focuses on the development of a BDJ color vision sensor, which then could provide input to a processing stage.

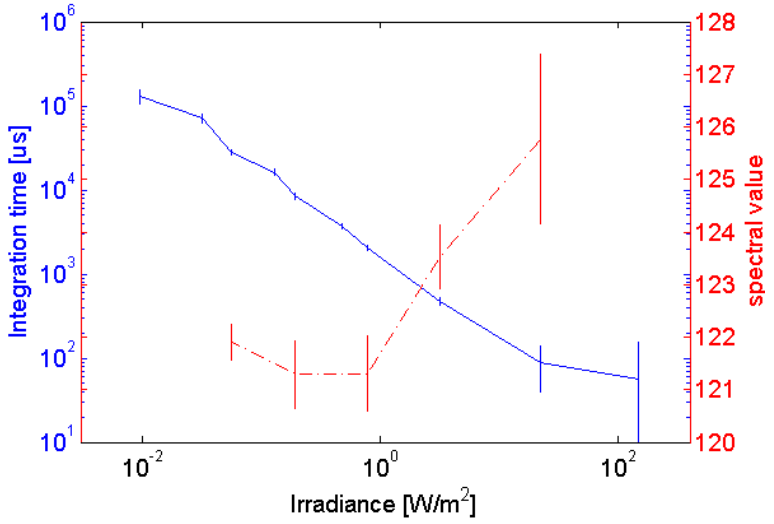


Figure 3.7: Integration time (blue, solid) and 8 bit color value (dashed, red) vs. irradiance.

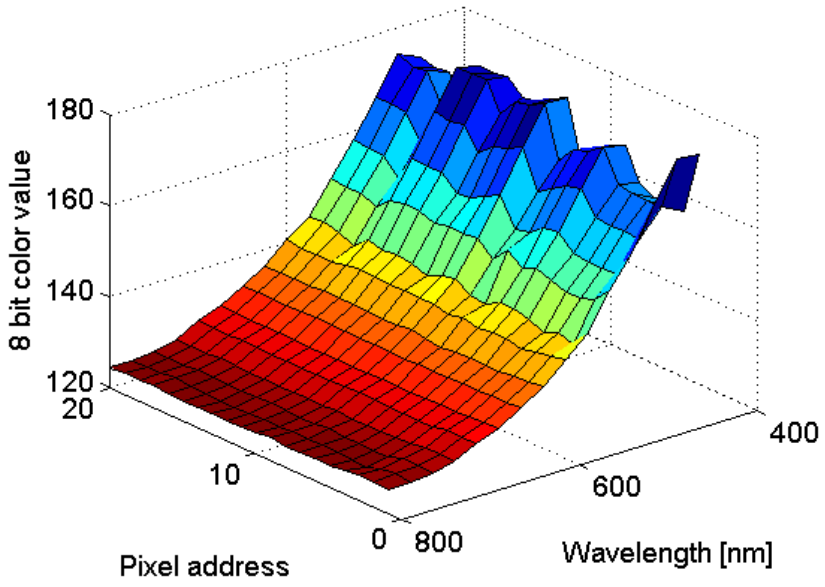


Figure 3.8: Spectral response of all 20 pixels vs wavelength.

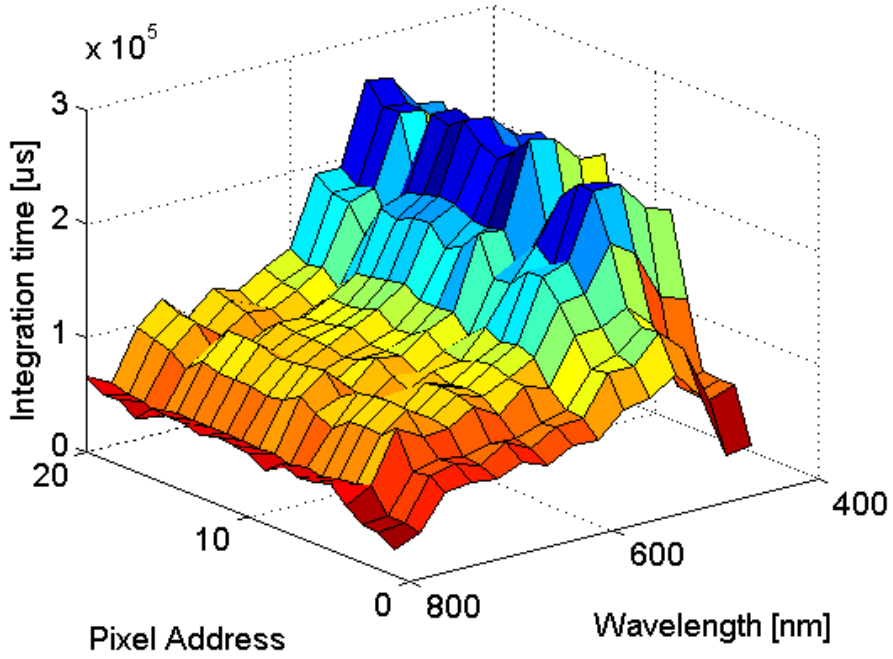


Figure 3.9: Integration time for all 20 pixels versus wavelength, normalized by irradiance. The integration time has been normalized by the irradiance measured with a Tektronix J17 photometer.

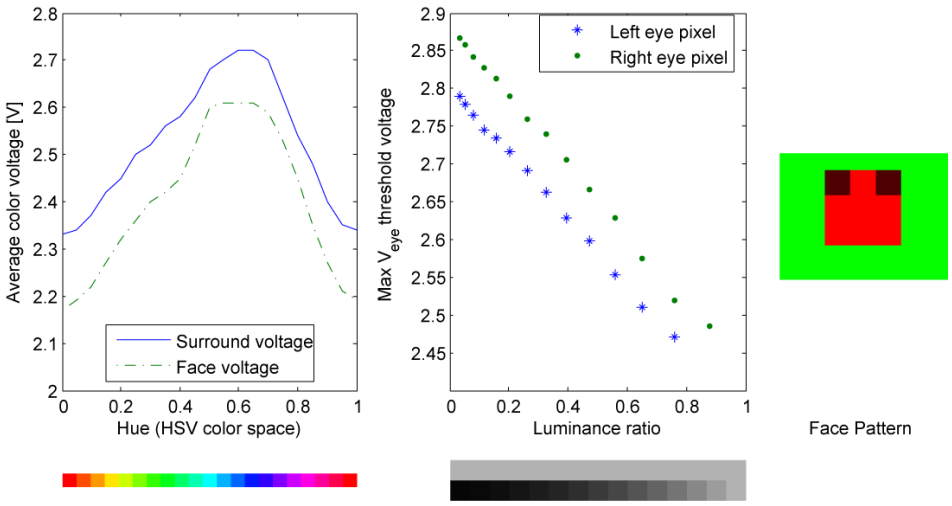


Figure 3.10: Pattern detector circuit response. Left plot shows the output of the capacitive averaging circuits. Right plot shows the maximum V_{eye} threshold voltage at which the sensor still can detect an eye for the given luminance ratio between eye pixel and surround. Reference voltages are $V_{Chi} = 2.985$ and $V_{Clo} = 2.463$ V. Rightmost the "face" stimulus pattern is shown.

Chapter 4

Novel event-based color change detection circuits

The previous chapter concluded with some reasoning why we shifted away from trying to develop a combined vision sensor/pattern-detector. In the rest of this thesis, we describe our efforts to develop the circuits necessary for building an event-based color silicon retina.

This chapter introduces novel pixel circuits which detect changes in wavelength, much as the DVS [66] detects changes in log intensity. The pixels employ buried double junctions (BDJ) to achieve dichromatic color sensitivity in standard CMOS. The pixels asynchronously emit events indicating whether the mean wavelength of the incident light has increased or decreased, i.e. whether the light has become 'bluer' or 'redder'.

4.1 The Pixel Circuits

A disadvantage of the BDJ is that only the currents I_T and $I_S = I_T + I_B$ are accessible for continuous time circuits, but not I_B directly. Fu and Titus [84] try to address this by subtracting a copy of I_T from I_S. This is problematic due to transistor mismatch and may work for a single pixel, but will cause very different pixel responses across an array.

As can be seen from Fig. 2.8, I_S/I_T varies almost a factor of five in the visible range for a $0.5\mu m$ process. We think this is sufficient for change detection, and therefore our pixel circuit works directly on I_T and I_S, which makes it more suitable for using in an array.

Four different pixel circuits are presented. The pixel circuits differ in the front-end and the way the ratio of the photocurrents is calculated; amplifiers, comparators and event generation are similar in all pixels. The gain of the switched capacitor amplifier is chosen differently for each pixel so that the overall pixel gain is the same. In simulation the output voltages of all the pixels respond with the same amplitude for a similar change in wavelength. Table 4.1 compares the pixel variants.

	LogSum	LogSum2	Current-mode	Current-mode 2
Photodiode Area	$690\mu m^2$	$690\mu m^2$	$690\mu m^2$	$690\mu m^2$
Number of transistors	29	23	40	39
Number of capacitors	6	6	5	5
Total pixel capacitance	1626.26 fF	1626.26 fF	783.2 fF	828.2 fF
Nominal Amplifier Gain	40.17dB	40.17dB	30.56dB	32.63dB
Number of gate biases	8	9	9	8
Number of source biases	2	2	5	4

Table 4.1: Color change pixel comparison

Where two opposite type gate biases for the same current are needed (V_{prBiasP} and V_{prBiasN}, V_{bufferP} and V_{bufferN}), the n-type bias is created from the p-type through a local mirror. These transistors are not shown in the pixel schematics but counted in the transistor count in table 4.1.

4.1.1 LogSum Pixel

The pixel circuit (Fig. 4.1) consists of a BDJ, two logarithmic current to voltage converters, a two-stage summing amplifier, two simple comparators and a reset and refractory circuit. Whenever the pixel crosses threshold, the summing amplifier is reset by closing the switches M_{r1} and M_{r2}, thereby memorizing the last value. The pixel computes the difference of the logarithms of I_S and I_T.

We now analyze the operation of the pixel starting from the logarithmic photoreceptor circuits. Because the summing amplifier discards DC input voltages, here we will only compute the changes in the photoreceptor outputs. The photoreceptors are biased to hold $V_{\text{sum}} > V_{\text{top}}$ so that each BDJ junction is reverse-biased.

Through the feedback transistor M_{N1} and the common source amplifier, consisting of M_{N2} and M_{P3}, the photodiode node V_{sum} is held at a nearly constant voltage. V_S settles at a voltage that makes M_{N1} source the photocurrent I_S.

The change in the n-type photoreceptor voltage V_S in response to a change in

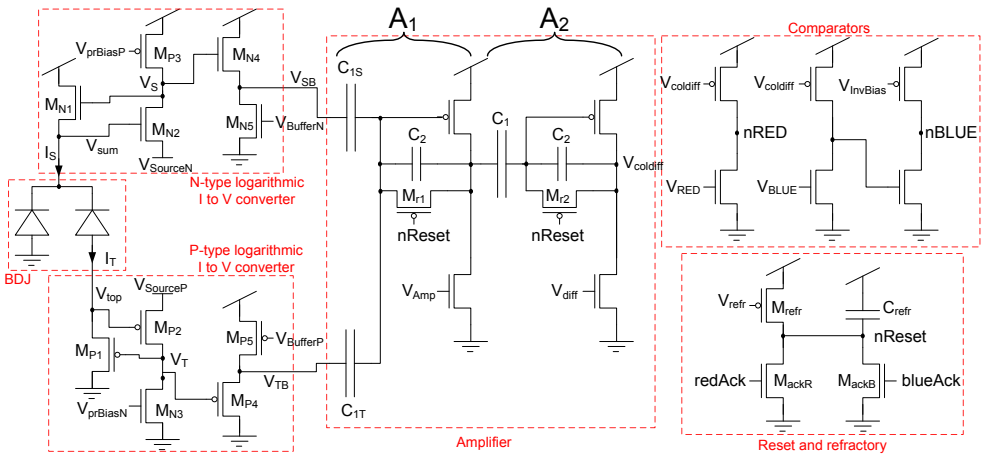


Figure 4.1: LogSum color change pixel circuit. V_{sourceN} and V_{sourceP} are source biases used to ensure proper biasing of the BDJ.

the summed BDJ current is

$$\Delta V_S = \Delta \frac{U_T}{\kappa_{N1}} \ln \frac{I_S}{I_{0N1}}, \tag{4.1}$$

where U_T is the thermal voltage kT/q, I_{0N1} the off-current of M_{N1}, and κ_{N1} the subthreshold back gate coefficient of M_{N1}. V_S is buffered by a source follower to produce V_{SB}:

$$\Delta V_{SB} = +\Delta U_T \frac{\kappa_{N4}}{\kappa_{N1}} \ln \frac{I_S}{I_{0N1}}. \tag{4.2}$$

The p-type photoreceptor works in a similar way, resulting in the source-follower output voltage V_{TB} in Eq. 4.3, which is opposite in sign to V_{SB}:

$$\Delta V_{TB} = -\Delta U_T \frac{\kappa_{P4}}{\kappa_{P1}} \ln \frac{I_T}{I_{0P1}}. \tag{4.3}$$

The source followers play two crucial roles: first, they buffer the photoreceptor outputs to drive the large capacitive loads of the summing amplifier. Second, they largely compensate the factor $1/\kappa$, so that $V_{TB} + V_{SB}$ is nearly independent of the difference between κ_n and κ_p.

The summing amplifier is implemented as two consecutive capacitive-feedback inverting amplifiers A_1 and A_2 [119], where A_1 has two summing input capacitances C_{1T} and C_{1S}. V_{amp} in A_1 is used to limit the bandwidth of the pixel for noise filtering and to balance possible bandwidth differences in the two photoreceptors. The gain of each stage is C_x/C_2, resulting in a total gain $A_T = C_{1T}C_1/C_2^2$ and $A_S = C_{1S}C_1/C_2^2$ respectively.

The output of the A_2 summing amplifier is its voltage change away from its reset value:

$$\Delta V_{\text{coldiff}} = U_T \left(A_S \frac{\kappa_{N4}}{\kappa_{N1}} \Delta \ln \frac{I_S}{I_{0N1}} - A_T \frac{\kappa_{P4}}{\kappa_{P1}} \Delta \ln \frac{I_T}{I_{0P1}} \right) \tag{4.4}$$

If we choose the ratio of A_S/A_T so that

$$A_S \frac{\kappa_{N4}}{\kappa_{N1}} = A_T \frac{\kappa_{P4}}{\kappa_{P1}} = A, \tag{4.5}$$

then the output simplifies to

$$\begin{aligned} \Delta V_{\text{coldiff}} &= A \cdot U_T \left(\Delta \ln \frac{I_S}{I_{0N1}} - \Delta \ln \frac{I_T}{I_{0P1}} \right) \\ &= A \cdot U_T \Delta \left(\ln \frac{I_S}{I_T} + \ln \frac{I_{0P1}}{I_{0N1}} \right) \\ &= A \cdot U_T \cdot \Delta \ln \left(\frac{I_S}{I_T}. \right) \end{aligned} \tag{4.6}$$

$\Delta V_{\text{coldiff}}$ responds only to changes in the ratio of the photo-currents. A change in overall intensity without a change in color does not change I_S/I_T.

The assumption that the response to intensity change is minimal when $A_S = A_T$ seems justified because the transistors M_{N1} and M_{N4} (respectively M_{P1} and M_{P4}) have the same geometry and the same gate-bulk voltage. Also, the ratio of the currents through these transistors $I_{M_{N1}}/I_{M_{N4}}$ and $I_{M_{P1}}/I_{M_{P4}}$ are similar because the photocurrents are of the same order of magnitude and the bias currents through M_{N4} and M_{P4} are the same. This means that the slopes of V_{TB} and V_{SB} versus photocurrent should be the same.

Due to the two stage amplifier, the junction leakage in the reset switch of the first stage is amplified by the second stage. Therefore we use C_1/C_2 parallel unit size switches in the second stage to approximately balance the leakage of the first stage.

The comparators compare V_{coldiff} against two thresholds V_{bluethr} and V_{redthr} set by V_{BLUE} and V_{RED}. The thresholds are offset from the reset level of V_{coldiff} to detect increasing and decreasing changes. V_{BLUE} and V_{RED} are adjusted so that after reset, the digital signals *nRED* and *nBLUE* are high. If the input of a comparator overcomes its threshold, a REDDER or BLUER event results. The output of the BLUE comparator is followed by a starved inverter to generate active low signals for both types of events.

These signals *nBLUE* and *nRED* are connected to the arbiter [76], and when the pixel is acknowledged, V_{nReset} is pulled to ground by M_{ackB} or M_{ackR}, thereby closing the reset switches M_{r1} and M_{r2}. The transistors M_{ackB} and M_{ackR} also enable an adjustable refractory period (implemented by M_{refr} and C_{refr}), which limits the maximum firing rate.

4.1.2 LogSum pixel, variant 2

The LogSum 2 pixel (Fig. 4.2) is very similar to the previous one, it also calculates the difference of the logarithms of I_S and I_T. It uses a different front-end [120] that does not have the factor $1/\kappa$ before the logarithm, therefore the gain should be

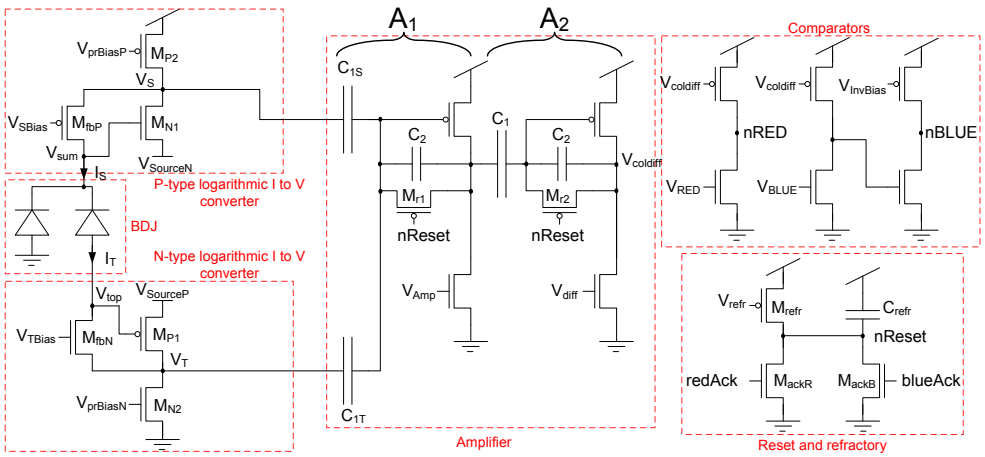


Figure 4.2: LogSum 2 color change pixel circuit. V_{sourceN} and V_{sourceP} are source biases used to ensure proper biasing of the BDJ.

better matched. However, to make use of this advantage, no source follower should be used to buffer V_S and V_T, as these would introduce a factor κ again.

In the following, we analyze the p-type logarithmic I-to-V converter. Through the feedback transistor M_{fbP}, the common source amplifier consisting of M_{N1} and M_{P2} holds the photodiode voltage V_{sum} nearly constant, V_S settles at a voltage so that transistor M_{fbP} supplies the photocurrent.

$$I_S = I_{0\text{fBp}} e^{\frac{1}{U_T}\left(\kappa_{\text{fBp}}(V_{dd}-V_{\text{SBias}})-(V_{dd}-V_S)\right)} \tag{4.7}$$

$$V_S = U_T \ln \frac{I_S}{I_{0\text{fBp}}} + V_{dd} - \kappa_{\text{fBp}}(V_{dd} - V_{\text{SBias}}) \tag{4.8}$$

The n-type front-end works similarly, changes in the photocurrent I_T result in

$$\Delta V_T = -U_T \Delta \ln \frac{I_T}{I_{0\text{fBn}}}. \tag{4.9}$$

Similar to the previous pixel, changes in the output voltage of the summing amplifier are then only proportional to the change in the ratio of the junction currents.

$$\Delta V_{\text{coldiff}} = A \cdot U_T \cdot \Delta \ln \frac{I_S}{I_T}. \tag{4.10}$$

Compared to the previous pixel, this circuit achieves the same gain using the same amount of capacitance and fewer transistors, but one more bias voltage. However, the forced omission of buffers between the front-end and the summing amplifiers could possibly be problematic.

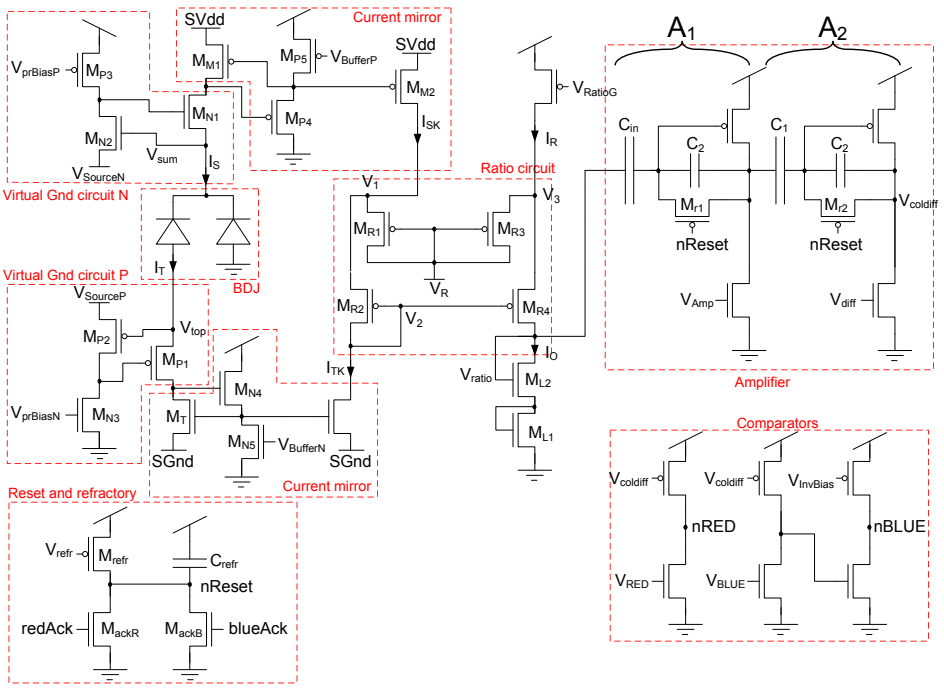


Figure 4.3: Current-mode pixel circuit. V_{sourceN} and V_{sourceP} are source biases used to ensure proper biasing of the BDJ. V_R is a source bias for the current-mode ratio circuit. *SGnd* and *SVdd* are shifted sources for sub-off-current copying [122].

4.1.3 Current-mode ratio circuit pixel

The current-model pixel circuit (Fig. 4.3) uses a current-mode circuit that calculates an output current proportional to the ratio of two input currents. This is a modification of a circuit presented by Patrick Merkli [121].

The pixel uses active front-ends similar to the LogSum pixel to hold the photodiodes at virtual ground. The photocurrents are copied using femto-ampere current mirrors [122] and then fed into the ratio-circuit. Transistors M_{L1} and M_{L2} are used to convert the output current of the ratio circuit into a voltage, which is fed to the capacitive amplifier. The rest of the circuit (second amplifier stage, comparators and reset/refractory period circuit) are the same as before.

Following is the analysis of the ratio circuit consisting of transistors M_{R1} to M_{R4}. The current through these transistors are

$$I_1 = I_{01} e^{\frac{\kappa(V_{dd}-V_R)-(V_{dd}-V_1)}{U_T}} \tag{4.11}$$

$$I_{TK} = I_{02} e^{\frac{\kappa(V_{dd}-V_2)-(V_{dd}-V_1)}{U_T}} \tag{4.12}$$

$$I_3 = I_{03} e^{\frac{\kappa(V_{dd}-V_R)-(V_{dd}-V_3)}{U_T}} \tag{4.13}$$

$$I_O = I_{04} e^{\frac{\kappa(V_{dd}-V_2)-(V_{dd}-V_3)}{U_T}} \tag{4.14}$$

Solving for I_O leads to

$$I_O = \frac{I_3 I_{TK}}{I_1} \frac{I_{01} I_{04}}{I_{02} I_{03}} \tag{4.15}$$

$$= \frac{(I_R - I_O) I_{TK}}{I_1} \frac{I_{01} I_{04}}{I_{02} I_{03}} \tag{4.16}$$

$$= \frac{(I_R - I_O) I_{TK}}{I_1} B \tag{4.17}$$

$$I_O = \frac{I_R I_{TK} B}{I_1 + I_{TK} B}. \tag{4.18}$$

In the absence of mismatch ($B = 1$, $I_{TK} = I_T$ and $I_{SK} = I_S$), I_1 is equal to I_B, so $I_1 + I_{TK} = I_S$ and the formula simplifies to

$$I_O = \frac{I_R I_T}{I_1 + I_T} = \frac{I_R I_T}{I_S}. \tag{4.19}$$

The transistors M_{L1} and M_{L2} convert the current I_O logarithmically into a voltage.

$$V_{\text{ratio}} = \frac{U_T}{\kappa_{L2}} (\ln \frac{I_O}{I_{0L2}} + \frac{1}{\kappa_{L1}} \frac{I_O}{I_{0L1}}) \tag{4.20}$$

For simplicity, we assume that $I_{0L1} = I_{0L2} = I_{0L}$.

$$V_{\text{ratio}} = U_T \frac{1 + 1/\kappa_{L1}}{\kappa_{L2}} \ln \frac{I_R I_T}{I_{0L} I_S} \tag{4.21}$$

The change in the voltage output of the capacitive amplifier is therefore

$$\Delta V_{\text{coldiff}} = U_T \frac{1 + 1/\kappa_{L1}}{\kappa_{L2}} A \Delta \ln \frac{I_R I_T}{I_{0L} I_S} \tag{4.22}$$

$$= U_T \frac{1 + 1/\kappa_{L1}}{\kappa_{L2}} A (\Delta \ln \frac{I_T}{I_S} + \Delta \ln \frac{I_R}{I_{0L}}) \tag{4.23}$$

$$\Delta V_{\text{coldiff}} = U_T \frac{1 + 1/\kappa_{L1}}{\kappa_{L2}} A \Delta \ln \frac{I_T}{I_S} \tag{4.24}$$

The gain of the capacitive amplifier is $A = C_{in} C_1 / C_2^2$. Because of the gain ($\frac{1+1/\kappa_{L1}}{\kappa_{L2}}$) provided by the transistors M_{L1} and M_{L2}, the capacitive amplifier gain A needs to be smaller compared to the two LogSum pixel variants to get the same response in V_{coldiff} for the same photocurrent ratio change.

The assumption of no mismatch is of course not valid and would make the response of several of these pixels across an array non-uniform. This pixel is therefore not suited for our purpose, but I realized this only after designing and fabricating this test chip.

4.1.4 Current-mode ratio circuit pixel 2

The current-mode 2 pixel circuit (Fig. 4.4) is similar to the previous current-mode pixel, but uses a different, translinear current mode circuit [92] to calculate the

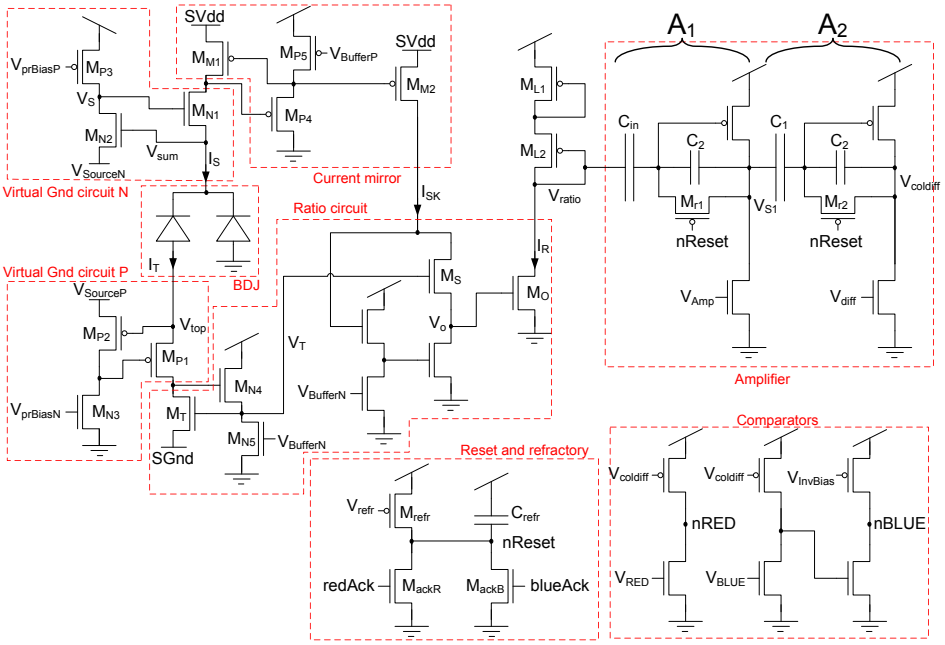


Figure 4.4: Current-mode 2 pixel circuit. V_{sourceN} and V_{sourceP} are source biases used to ensure proper biasing of the BDJ. *SGnd* and *SVdd* are shifted sources for sub-off-current copying [122].

ratio between the photocurrents. The circuit is less sensitive to mismatch, more compact and needs less biases, but has slightly less gain.

To analyze the ratio circuit, we calculate the current through transistors M_T, M_S and M_O:

$$I_T = I_{0T} e^{\frac{\kappa_T V_T - SGnd}{U_T}} \tag{4.25}$$

$$I_{SK} = I_{0S} e^{\frac{\kappa_S V_T - V_o}{U_T}} \tag{4.26}$$

$$I_O = I_{0O} e^{\frac{\kappa_O V_o}{U_T}} \tag{4.27}$$

From these equations, we derive

$$\frac{V_T}{U_T} = \frac{\ln \frac{I_T}{I_{0T}} + \frac{SGnd}{U_T}}{\kappa_T} \tag{4.28}$$

$$\frac{V_O}{U_T} = \frac{\kappa_S}{\kappa_T} \ln \frac{I_T}{I_{0T}} - \ln \frac{I_{0S}}{I_{SK}} + \frac{SGnd}{U_T} \frac{\kappa_S}{\kappa_T} \tag{4.29}$$

$$\frac{V_O}{U_T} = \ln \left(\left(\frac{I_T}{I_{0T}} \right)^{\kappa_S/\kappa_T} \frac{I_{0S}}{I_{SK}} \right) + \frac{SGnd}{U_T} \frac{\kappa_S}{\kappa_T} \tag{4.30}$$

From this, we can derive

$$I_O = I_{0O} e^{\frac{\kappa_O \kappa_S}{\kappa_T} \frac{SGnd}{U_T}} \left(\frac{I_T^{\kappa_S/\kappa_T}}{I_{SK}} \frac{I_{0S}}{I_{0T}^{\kappa_S/\kappa_T}} \right)^{\kappa_O} \tag{4.31}$$

For a correct response, κ_S has to be equal to κ_T. This assumption is much more justifiable than the assumption of no mismatch in the off-current in the current-mode pixel, because mismatch in κ is much smaller than mismatch in the off-current.

This simplification leads to

$$I_O = I_{0O} e^{\frac{\kappa_O SGnd}{U_T}} \left(\frac{I_T}{I_{SK}} \frac{I_{0S}}{I_{0T}} \right)^{\kappa_O} . \tag{4.32}$$

Similar to the previous pixel, M_{L1} and M_{L2} convert the output current logarithmically to a voltage, the change in the output voltage of is thus

$$\Delta V_{\text{coldiff}} = A \kappa_O U_T \frac{1 + 1/\kappa_{L1}}{\kappa_{L2}} \Delta \ln \frac{I_T}{I_{SK}} \tag{4.33}$$

Mismatch in the current mirror from I_S to I_{SK} is not critical in this pixel, because mismatch leads only to a different DC level of V_{ratio}. This does not matter, because we are only interested in changes.

4.2 ColTmpDiff test chip

A test chip was fabricated through the MOSIS service in an AMI $0.5\mu m$ 3M 2P process with a supply voltage of 5V, die size is $2.2 \times 2.2 mm^2$. Fig 4.5 shows a die photo and a close-up of the LogSum pixel layout. The test chip includes a spectral measurement circuit as described in 2.4.2, the four different test pixel designs described above, a new sensitive DVS pixel [80] and a DVS pixel with two-stage amplification.

4.3 Measurement Results

For the measurements, we stimulated the pixels with a red and a blue LED. The measurement setup is described in Appendix B. To measure the response to intensity changes, the red LED was stimulated alone, while the blue LED was off. To measure the response to a light color change, we stimulated the pixel with the two LEDs simultaneously, while varying the ratio of the currents through the LEDs.

Because the number of carriers generated from photons at a given wavelength is linearly proportional to the light intensity at this wavelength, superposition of two light sources at two different wavelengths evokes the same photocurrent as stimulation with a monochromatic light source with an intensity that is calculated as a weighted sum of the intensities of the two light sources. The weights follow from the spectral sensitivity curve of the semiconductor.

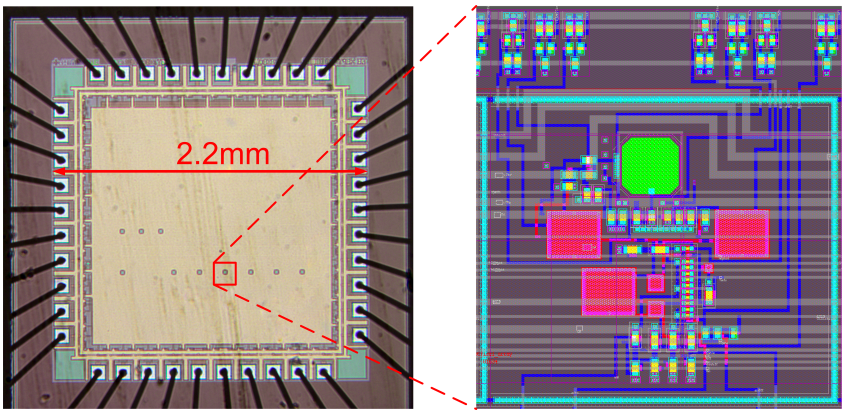


Figure 4.5: ColTmpDiff die photo and the LogSum pixel circuit layout with local buffers. Metal 3 is used to shield the circuits from light. In the die photo, the openings in metal 3 for the photodiodes can be seen. The layout of the circuits was not optimized for area.

4.3.1 DC response

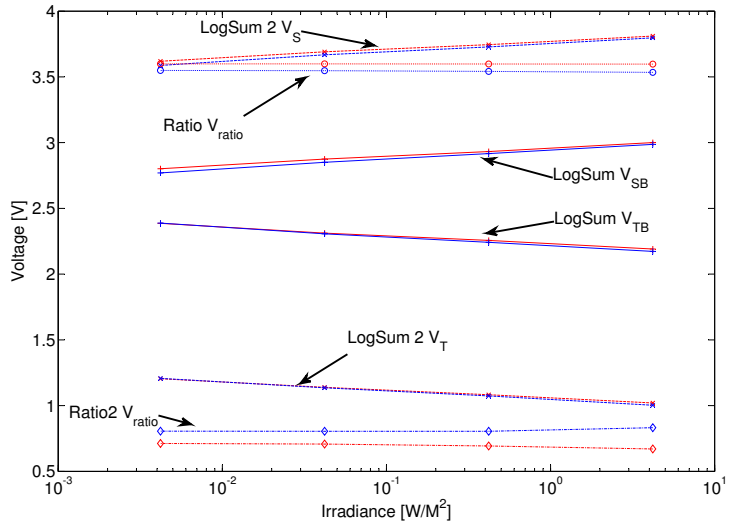
First, the DC behavior of the pixel circuits is investigated. Fig. 4.6(a) shows the photoreceptor output voltages of the LogSum pixels and the ratio circuit output voltages of the current-mode circuits for three decades of illumination and two different light colors.

For all pixels, the input to the differentiator should be independent of the illumination level for constant color. For the LogSum pixels, this input consists of the sum of the photoreceptor output voltages, for the current-mode circuits the input to the differentiator is the ratio circuit output voltage. Fig. 4.6(b) shows the inputs to the differentiator with the DC component (arbitrarily defined as the response to blue light at the highest measured intensity) removed.

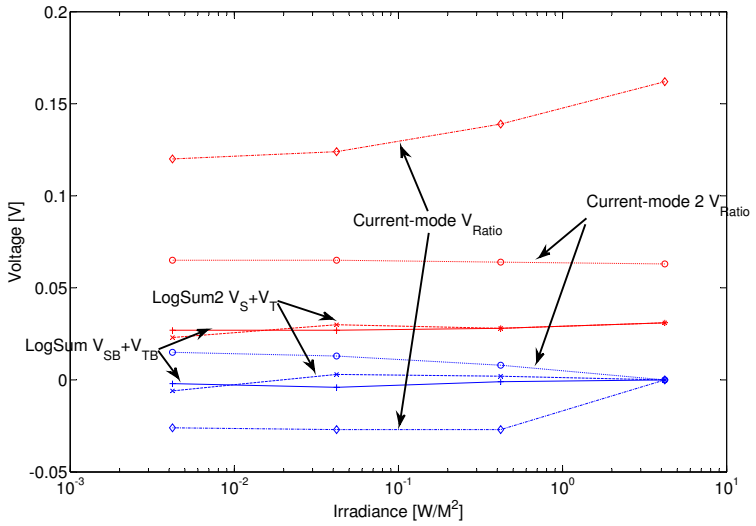
For all the pixels the response to blue light is clearly distinguishable from the response to red light. As expected, the current-mode circuits have higher gain, and the gain of the two different LogSum pixel variants is very similar. All the pixels show some unwanted response to illumination change. Interesting is the ratio between the response to color change and the response to intensity change, which is shown in table 4.2. The LogSum pixel and the current-mode 2 pixel are more robust to intensity changes.

4.3.2 Response to step inputs

Next the response of the pixels to step inputs of changing intensity and constant color, as well as changing color and intensity have been evaluated. For a step intensity change, the chip is stimulated with the red LED, which is driven by a square current changing by a factor two. To produce a color change, the blue LED is added with a constant current, thus the ratio between the LED currents changes.



(a) DC voltages



(b) Differentiator input with DC component removed

Figure 4.6: DC response of the four pixel variants to blue and red light at different intensities. Blue curves are for stimulation with the blue LED, red curves for the red LED.

Pixel	Color response/intensity response
LogSum	18
LogSum 2	7.1
Current-mode	10
Current-mode 2	19.5

Table 4.2: Ratio of mean response to color change from red to blue and mean response to intensity change of one decade.

Intensity changes from $6.7W/m^2$ to $3.35W/m^2$ for the intensity change stimulus and from $10.72W/m^2$ to $7.4W/m^2$ for the color and intensity change stimulus. Unfortunately we do not have means to measure the spectral content of the light directly. We estimated the mean wavelength change by means of the spectral sensitivity measurement circuit (see section 2.4.2). The mean wavelength changes roughly from 555nm to 525nm.

Figures 4.7 to 4.10 show traces of V_{coldiff} at different illumination levels for all the pixels. The illumination level has been changed by means of Kodak Wratten neutral density filters. Event thresholds are set very high to prevent the pixels from spiking. The bias V_{amp} has been adjusted so that the bandwidth at V_{coldiff} in all the pixels is roughly 100Hz.

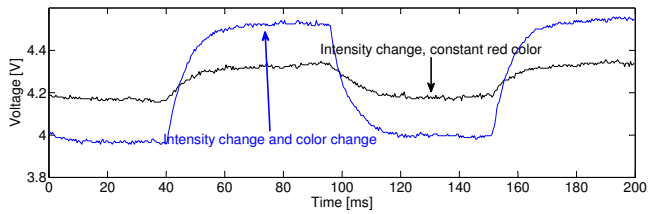
All the pixels respond more strongly to color change than to intensity change only. For all pixels the response at the highest illumination level is stronger than the response at lower intensities. This could be because some transistors (the feedback transistors in the LogSum pixels, the ratio-circuit transistors in the current-mode pixels) are working in moderate inversion instead of weak inversion and thus the transistors voltage to current relation is not exponential.

Three of the four pixels show a nice response to color change over three decades of illumination, but at the lowest illumination level (factor 1000 attenuation), the traces become noisy. At even lower illumination, the pixels are unusable due to even higher noise level.

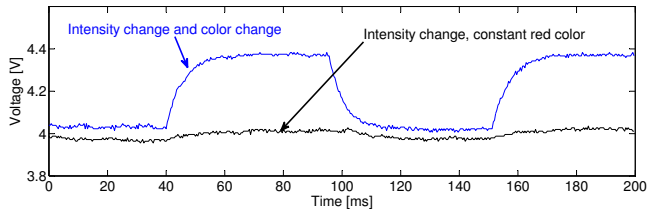
As predicted from the DC response curve, the current-mode pixel responds strongly to intensity change. Interestingly, the sign of the response to intensity change versus the input flips from the stimulation with no filter (Fig. 4.9(a)) to the stimulation attenuated with a ND1 filter (Fig. 4.9(b)).

At the lowest illumination level, the current-mode pixel also shows a very pronounced band-pass-like behavior. Even though it looks like the bandwidth of V_{coldiff} increased compared to higher illumination, the bias V_{amp} has not been changed. The amplitude of the bandpass response can be reduced with the V_{amp} bias voltage at the expense of reducing pixel bandwidth also for desired responses to color change.

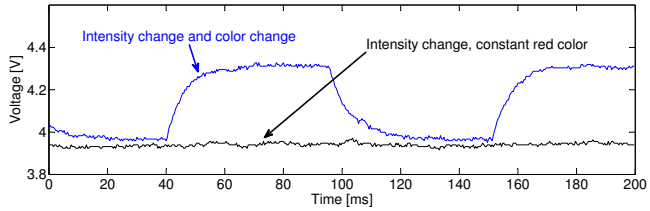
The reasons for the phenomena described in the last two paragraphs were not investigated, because the current-mode pixel is anyway not suited for further development due to its sensitivity to mismatch, as described in section 4.1.3.



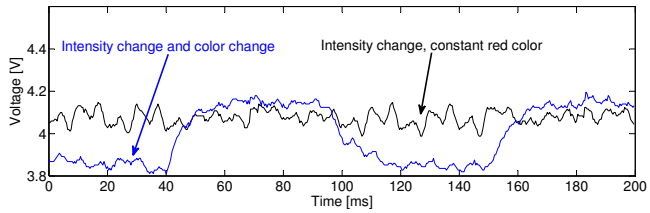
(a) No filter



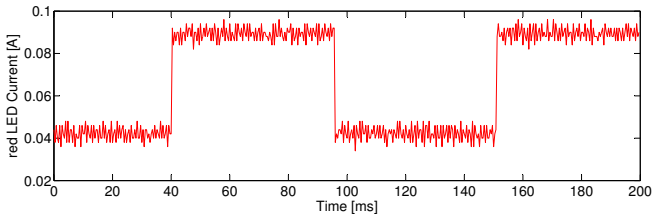
(b) ND1 filter (factor 10 attenuation)



(c) ND2 filter (factor 100 attenuation)

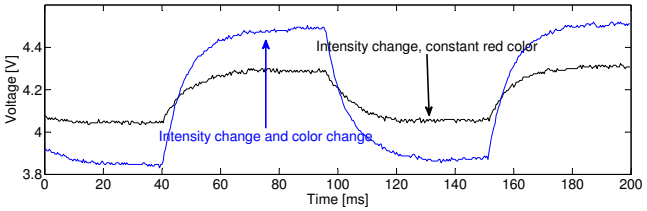


(d) ND3 filter (factor 1000 attenuation)

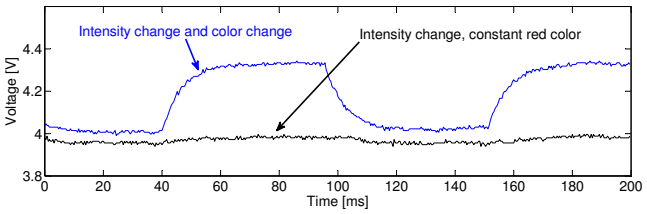


(e) red LED driving current

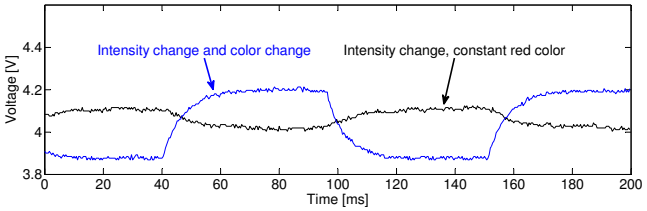
Figure 4.7: LogSum pixel, response to step input of intensity change (one octave) with constant color, and intensity and color change. For intensity change only, the chip is stimulated with the red LED only, the blue LED is turned off. For color change, the blue LED is additionally turned on with a constant current.



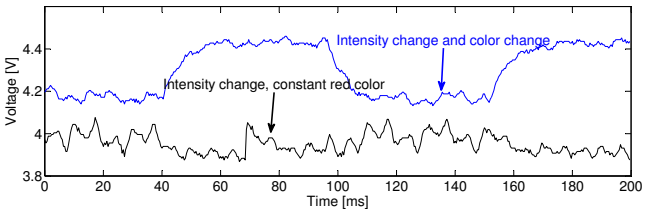
(a) No filter



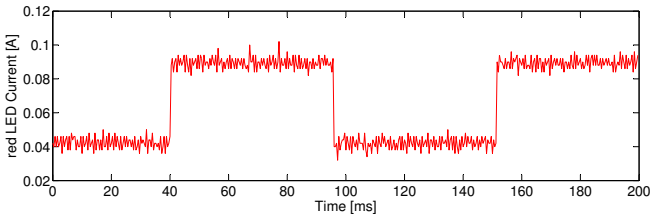
(b) ND1 filter (factor 10 attenuation)



(c) ND2 filter (factor 100 attenuation)

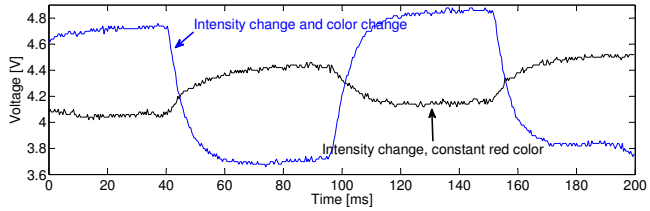


(d) ND3 filter (factor 1000 attenuation)

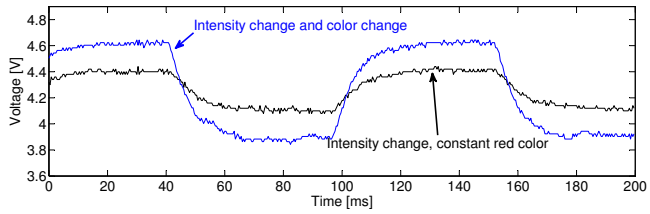


(e) red LED driving current

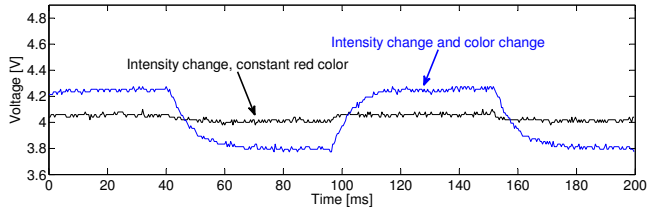
Figure 4.8: LogSum2 pixel, response to step input of intensity change (one octave) with constant color, and intensity and color change. For intensity change only, the chip is stimulated with the red LED only, the blue LED is turned off. For color change, the blue LED is additionally turned on with a constant current.



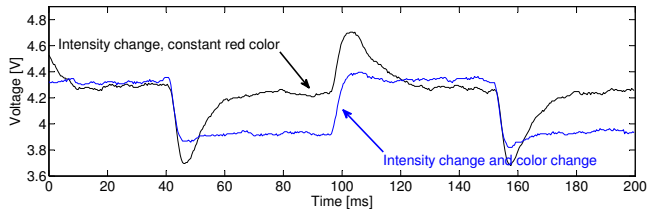
(a) No filter



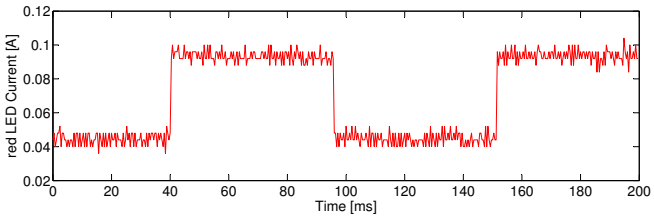
(b) ND1 filter (factor 10 attenuation)



(c) ND2 filter (factor 100 attenuation)

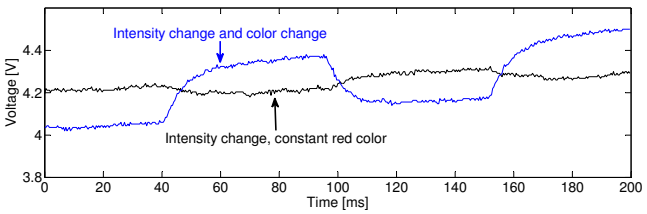


(d) ND3 filter (factor 1000 attenuation)

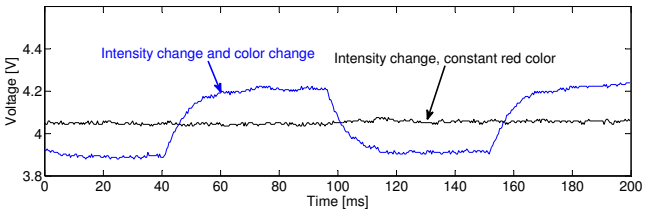


(e) red LED driving current

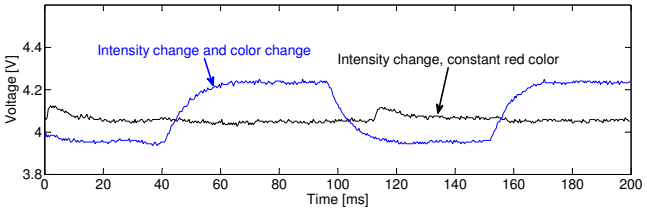
Figure 4.9: Current-mode pixel, response to step input of intensity change (one octave) with constant color, and intensity and color change. For intensity change only, the chip is stimulated with the red LED only, the blue LED is turned off. For color change, the blue LED is additionally turned on with a constant current. Note that the voltage scale is different compared to the plots of the other pixel variants.



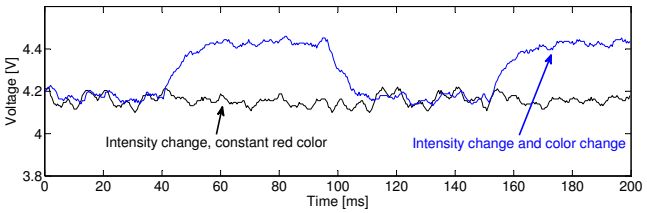
(a) No filter



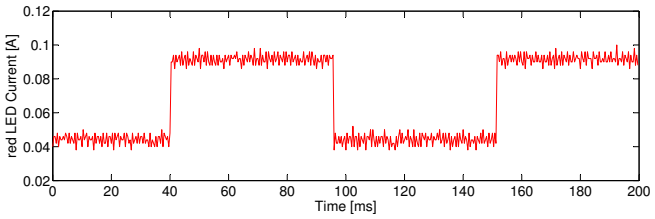
(b) ND1 filter (factor 10 attenuation)



(c) ND2 filter (factor 100 attenuation)



(d) ND3 filter (factor 1000 attenuation)



(e) red LED driving current

Figure 4.10: Current-mode 2 pixel, response to step input of intensity change (one octave) with constant color, and intensity and color change. For intensity change only, the chip is stimulated with the red LED only, the blue LED is turned off. For color change, the blue LED is additionally turned on with a constant current.

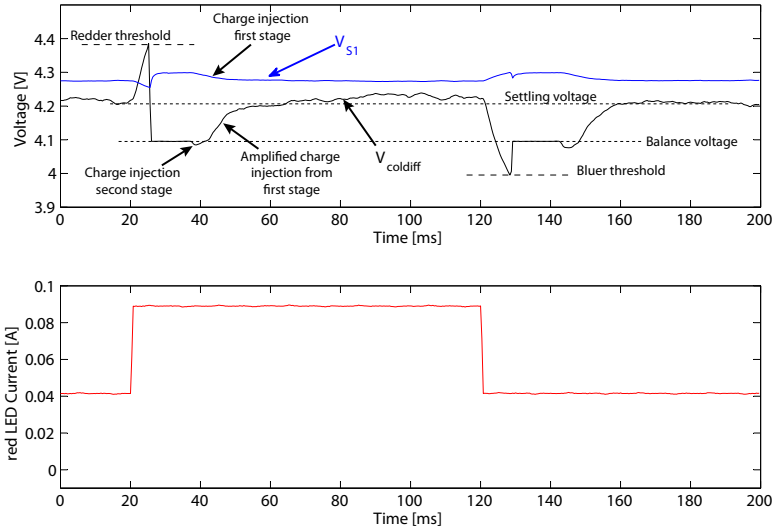


Figure 4.11: Charge injection from the reset switches in the current-mode 2 pixel after an event due to a step color change.

The V_{coldiff} curves of the the current-mode 2 pixel are more strongly tilted for stimulation with highest intensity (Fig. 4.10(a)). The tilt is gone for attenuated light, thus we can conclude that is due to parasitic photocurrents in the differentiator reset switch M_{r1}, even though this switch is completely covered by metal.

In the next measurement, we adjusted the thresholds so that the pixels could generate color change events. This is illustrated in Fig. 4.11, showing the response of V_{coldiff} of the current-mode 2 pixel to a step color change similar to the previous experiments. After the step increase of the LED current, V_{coldiff} rises to the REDDER threshold level and is then reset.

After resetting the differentiator, the V_{coldiff} trace shows strong charge injection from opening the reset switches M_{r1} and M_{r2}. The problem is that the charge injection of the first stage is amplified by the second stage. When the reset voltage *nReset* (not shown, this node is not instrumented) slews back to V_{DD}, it crosses V_{coldiff} before crossing V_{S1}, thus the second stage of the amplifier comes out of reset before the first stage. When the first stage comes out of reset, charge from the channel in M_{r1} is injected into the floating node of A_1, causing V_{S1} to decrease a bit. This charge injection is then amplified by the second amplifier stage. Thus the reset circuit has to be improved by using separate signals for M_{r1} and M_{r2} to avoid the amplification of the charge injection due to M_{r1}, by holding the second stage slightly longer in reset than the first stage.

This problem of charge injection is present in all four pixel variants, the traces of the LogSum, the LogSum2 and the current-mode pixels are very similar to the traces of the current-mode 2 pixel shown in Fig. 4.11. The problem did not show

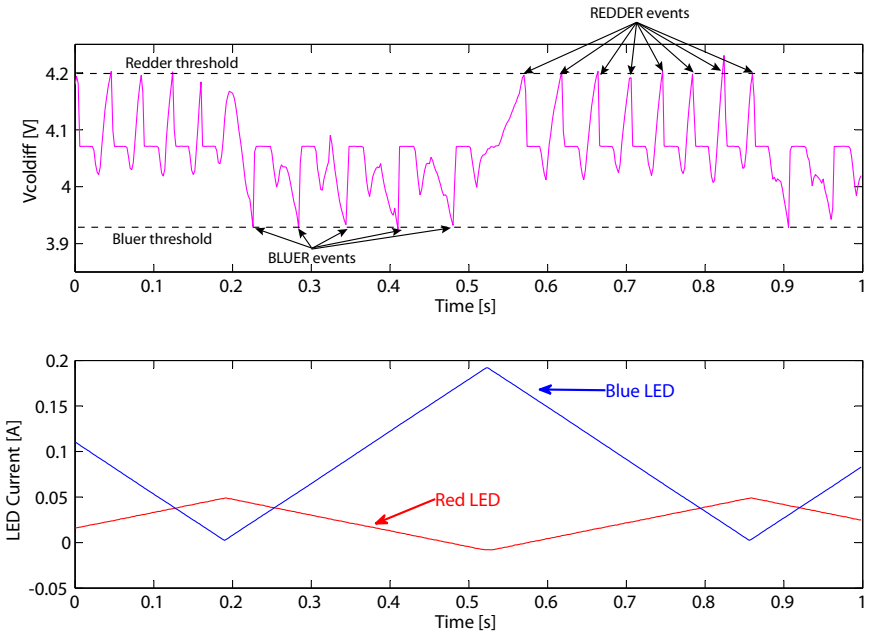


Figure 4.12: LogSum pixel color sensitivity

up to this extent in Spice simulations.

For the following measurements, we had to set the refractory bias V_{refr} quite close to V_{DD} so that the switches close slowly, which reduces the effect of charge injection.

4.3.3 Color sensitivity

To characterize the color sensitivity, we counted the number of events the pixels produce for a color sweep from blue to red and vice versa. The LEDs are stimulated with 1.5Hz 180° out-of-phase triangular currents (illustrated for example in Fig. 4.12), so that the intensity stayed approximately constant. The light was attenuated with a ND1 filter.

Due to the charge injection, especially the BLUER threshold (REDDER for the current-mode pixel) can not be set close to the settling voltage. This results in a reduced number of events per sweep and an asymmetric response. Improving the reset circuit to reduce the charge injection will allow lower threshold and higher color sensitivity and a more symmetric response. It will also allow to increase the refractory bias current and shorten the refractory time.

The current-mode pixel produces the most events for a color sweep, 8 REDDER and 16 BLUER events per sweep. This amounts to a color sensitivity of 22.5nm for increasing wavelength and 11.25nm for decreasing wavelength. Both LogSum and LogSum 2 pixels generate between 5 to 6 BLUER and 8 to 9 REDDER events per

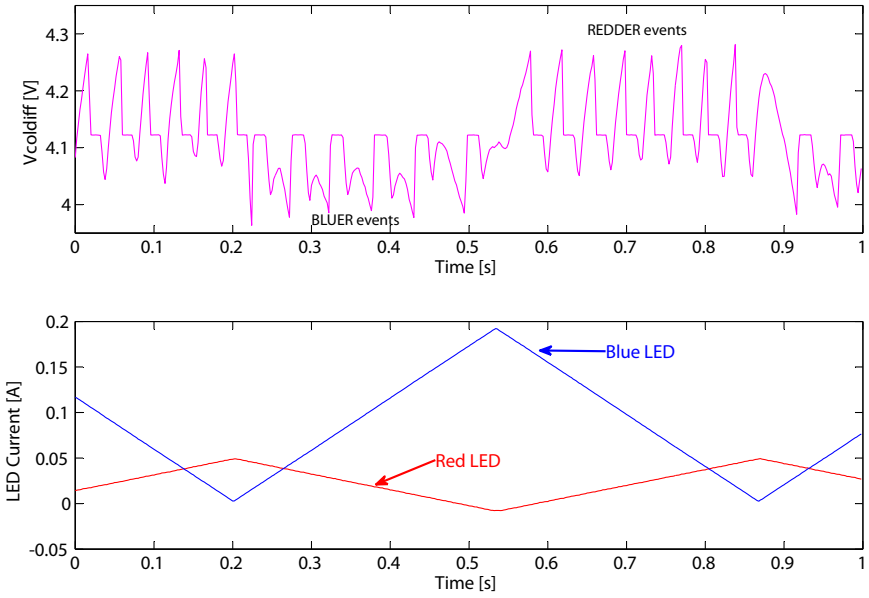


Figure 4.13: LogSum 2 pixel color sensitivity

sweep, which amounts to around 30nm and 20nm wavelength change sensitivity.

4.4 Discussion

This chapter demonstrates successfully how a BJD can be used to create event-based color change detection pixels. Four different pixel circuits are presented, the measurement results show that the LogSum pixel and the current-mode 2 pixel circuit are most suitable for further investigation, because they show a better ratio of the response to color change versus response to intensity change.

The color change sensitivity is around 20 to 30nm, it is limited by a flawed reset circuit. This flawed reset circuit is a common problem of all the pixel circuits and it results in strong charge injection after resetting the differentiator. The reset circuit has to be improved by using separate signals for the reset switches M_{r1} and M_{r2} to avoid the amplification of the charge injection due to M_{r1}, by holding the second amplifier stage slightly longer in reset than the first stage.

Compared to the Logsum pixel, the LogSum 2 pixel needs one more bias voltage but less transistors. A disadvantage is that due to the forced omission of source follower buffers, this pixel circuit is not suited for combination of color change detection and intensity change detection, because the voltage bounces created by resetting either event pathway would likely interfere with the other event pathway and generate bogus events.

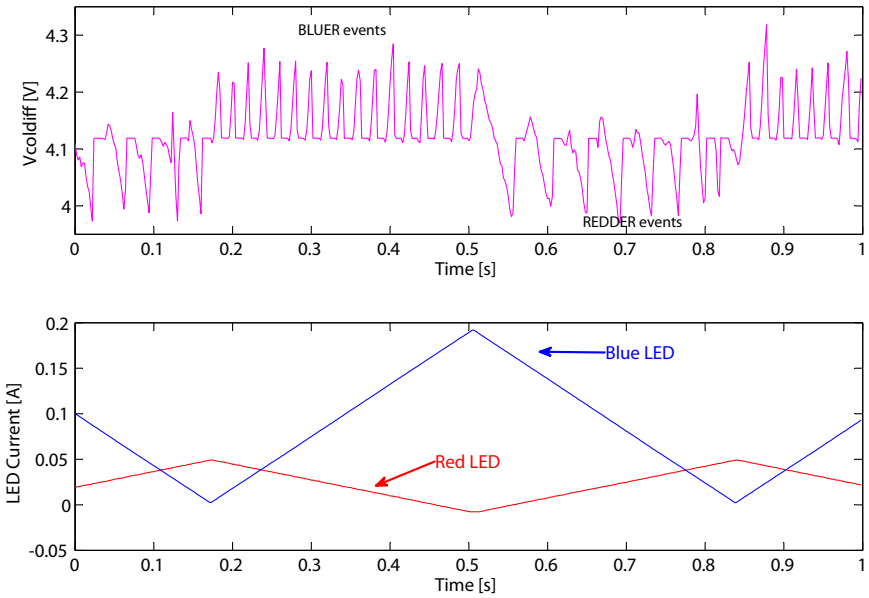


Figure 4.14: Current-mode pixel color sensitivity

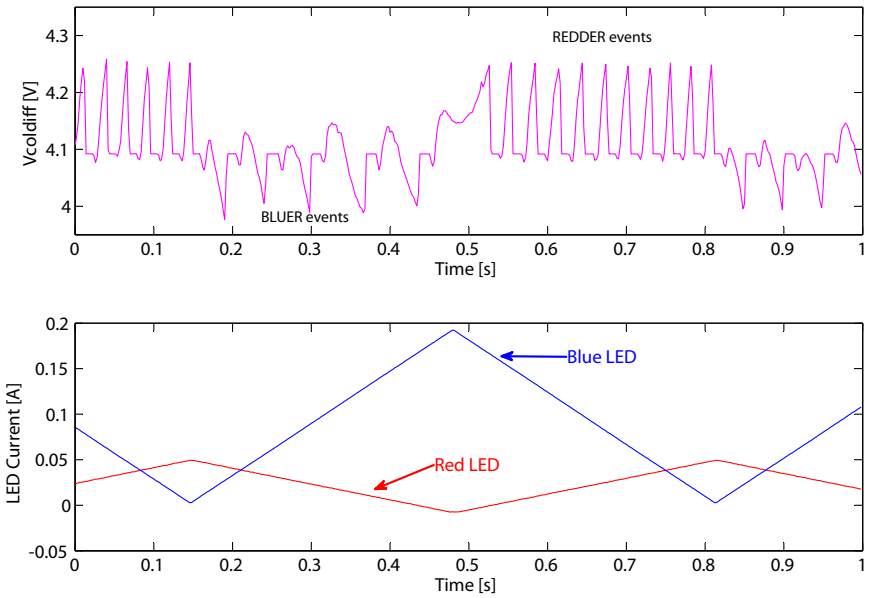


Figure 4.15: Current-mode 2 pixel color sensitivity

The current-mode pixel is not suitable for further development due to different reasons: detailed analysis of the current-mode pixel revealed its susceptibility to mismatch, it is the most complex pixel, needs the highest number of bias voltages and measurements show a relatively strong response to change of illumination.

Both LogSum and current-mode 2 pixels can easily be combined with an intensity change detection pathway based on the DVS pixel [66] and share the same photodiode and parts of the circuit. In the LogSum pixel, the intensity pathway can use an additional source follower connected to V_S to track changes of intensity. In the current-mode 2 pixel, the output V_S of the photoreceptor used to keep V_{sum} at a virtual ground can be used to track the intensity.

Most modern mixed-mode process technologies – including the UMC 180nm process used for the succeeding chips – offer MiM (metal-insulator-metal) capacitors, which are stacked above the circuits. This allows to put a substantial amount of capacitance in a pixel without sacrificing area (in the DVS pixel, which is built in a process technology that does not offer MiM capacitors, more than 50% of the pixel area is used for the capacitors). This favors the LogSum pixel over the current-mode 2 pixel due to the lower transistor count. For implementation in process without MiM capacitors, the current-mode 2 pixel allows smaller implementation due to lower pixel capacitance.

Based on our decision to fabricate succeeding chips in the "UMC L180 Mixed-Mode/RF" technology with MiM capacitors, the LogSum pixel has been chosen for further development. Chapter 5 shows a detailed small-signal analysis of this pixel circuit, chapter 6 shows the combination of this pixel circuit with an intensity change detection pathway and chapter 8 describes the implementation of a small array which is based on this pixel.

Chapter 5

Detailed color pixel analysis

Chapter 4 introduced novel pixel circuits that detect changes of wavelength and do not react to changes of illumination with constant color. The Log-Sum pixel presented in subsection 4.1.1 appears to be the most promising circuit. In this chapter, we analyze the small signal behavior of the Log-Sum color pixel front-end. In section 5.1, we derive the small signal transfer function to investigate AC signal performance and bandwidth of the circuit. Section 5.2 investigates the effect of a cascode transistor in the common source amplifier and section 5.3 calculates expected noise. In the whole analysis, capital letters I_x and V_x denote operating point currents and voltages, while i_x and v_x denote small deviations from the operating points.

5.1 Small-signal transfer function of the front-end

Fig. 5.1 shows the color pixel front-end and its corresponding small-signal model. Most of the drain-source conductances have been ignored, because they are significantly lower than their corresponding transconductances. The gate-source capacitance of M_{ampP} would introduce another pole, but its capacitance is much smaller than the parasitic photodiode capacitance, and thus its pole is at a much higher frequency.

The transconductances g_{ma} of the amplifier transistors M_{ampN} and M_{ampP} are assumed to be equal. In reality, they will not be exactly equal because their κ is going to be different even though they carry the same bias current and have equal geometry. C_m is the Miller capacitance, consisting of the gate-drain overlap capacitance of M_{ampN} and the gate-source overlap capacitance of M_{fbN}, and M_{ampP} and M_{fbP} respectively. It is assumed to be equal for both photoreceptors.

The following equations derive Kirchhoff's current law at the nodes v_w, v_a, v_t and v_s.

$$i_b + i_t = g_{mfn}(v_s - v_w) - sC_b v_w - sC_u(v_w - v_a) + sC_m(v_s - v_w) \tag{5.1}$$

$$i_t = g_{mfp}(v_a - v_t) - sC_u(v_w - v_a) + sC_m(v_a - v_t) \tag{5.2}$$

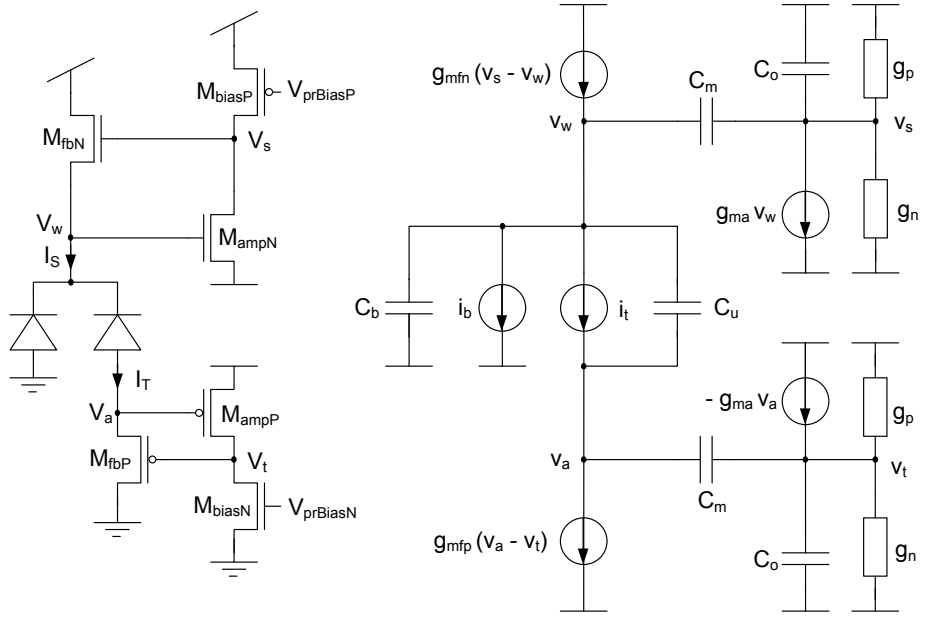


Figure 5.1: cDVS front-end and corresponding small-signal model

$$0 = g_{ma}v_a + g_o v_t + sC_m(v_t - v_a) + sC_o v_t \tag{5.3}$$
$$0 = g_{ma}v_w + g_o v_s + sC_m(v_s - v_w) + sC_o v_s, \tag{5.4}$$

The output conductance g_o consists of the drain-source conductances of the amplifier and the bias transistor in parallel, i.e. $g_o = g_n + g_p$. This value will be very similar for both photoreceptors because the transistors have the same geometry and the same current flowing through them. $g_{mfn} = \kappa_n I_s/U_T$ and $g_{mfp} = \kappa_n I_t/U_T$ are the transconductances of the feedback transistors M_{fbN} and M_{fbP}.

We define the following time constants, as well as the amplifier gain

$$\begin{aligned} \tau_m &= \frac{C_m}{g_{ma}}, \quad \tau_n = \frac{C_m}{g_{mfn}}, \quad \tau_p = \frac{C_m}{g_{mfp}} \\ \tau_b &= \frac{C_b}{g_{mfn}}, \quad \tau_u = \frac{C_u}{g_{mfp}}, \quad \tau_w = \frac{C_u + C_b}{g_{mfn}} \\ \tau_t^2 &= \frac{C_u^2}{g_{mfn}g_{mfp}}, \quad \tau_o = \frac{C_o}{g_{ma}}, \quad A = \frac{g_{ma}}{g_o} \end{aligned} \tag{5.5}$$

From equations 5.3 and 5.4, we can derive

$$\frac{v_a}{v_t} = \frac{v_w}{v_s} = -\frac{1}{A}\frac{1 + sA(\tau_m + \tau_o)}{1 - \tau_m s} = -K(s). \tag{5.6}$$

Using this in equation 5.2, we solve for

$$v_t = \frac{-i_t/g_{mfp} + s\tau_u K v_s}{(1+K)(1+s\tau_p) + sK\tau_u}. \tag{5.7}$$

Entering this into equation 5.1 and using 5.6, we can solve for the transfer function from the current sources to v_s and v_t:

$$\frac{v_s}{i_s} = \frac{\frac{1}{g_{mfn}}\left((1+K)(1+s\tau_p) + sK\tau_u - sK\tau_u\frac{i_t}{i_s}\right)}{((1+K)(1+s\tau_n) + sK\tau_w)((1+K)(1+s\tau_p) + sK\tau_u) - s^2\tau_t^2K^2} \tag{5.8}$$

$$\frac{v_t}{i_t} = \frac{-\frac{1}{g_{mfp}}\left((1+K)(1+s\tau_n) + sK\tau_b - sK\frac{C_u}{g_{mfn}}\frac{i_b}{i_t}\right)}{((1+K)(1+s\tau_n) + sK\tau_w)((1+K)(1+s\tau_p) + sK\tau_u) - s^2\tau_t^2K^2}, \tag{5.9}$$

where $i_s = i_b + i_t$. In both nominators, we see some terms proportional to the ratio of the photocurrents, which is unwanted. These terms are due to the coupling through C_u. To minimize the coupling relative to the signal, we should maximize the amplifier gain A.

The DC gain, given as the response to a relative current change in units of the thermal voltage, is

$$\frac{v_s/U_T}{i_s/I_s} = \frac{1}{\kappa_n}\frac{1}{1+\frac{1}{A}} \tag{5.10}$$

$$\frac{v_t/U_T}{i_t/I_t} = -\frac{1}{\kappa_p}\frac{1}{1+\frac{1}{A}}. \tag{5.11}$$

Fig. 5.2 shows bode plots of these two transfer functions for different photocurrent levels and small signal parameters taken from a Spice simulation of the pixel presented in chapter 6, along with the Spice AC simulation results. The curves show a good agreement over the whole photocurrent range up to 100MHz, where additional parasitic capacitances start to influence. For high photocurrents (>1pA), the time-constants of the photodiode nodes and the feedback amplifier start to be in a similar range and thus the systems is ringing. In order to avoid the ringing, the amplifier has to be made faster by increasing the bias current. Reasonable bandwidth is only achieved for photocurrents bigger than hundred femtoamperes.

The transfer functions are of fourth order and are very difficult to analytically analyze. We therefore analyze the circuit under two different assumptions.

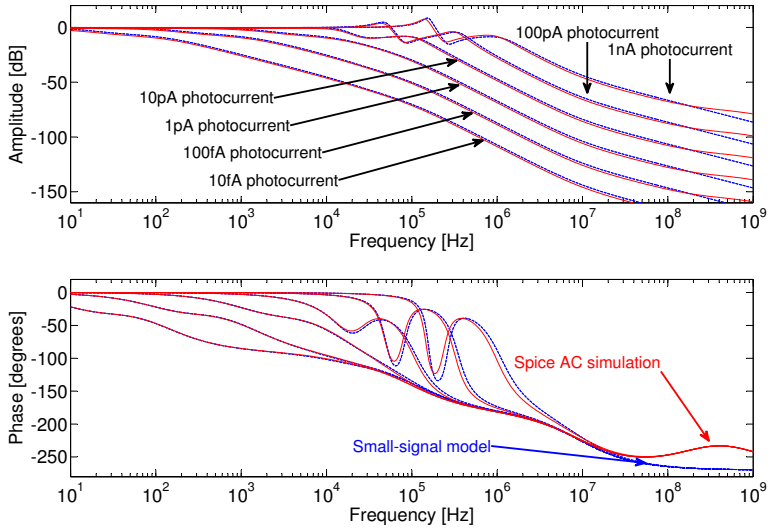
5.1.1 Assumption 1: infinitely fast amplifier

If the amplifiers are sufficiently fast (i.e $A\tau_o, A\tau_m \ll \tau_w, \tau_u, \tau_n, \tau_p$), we can approximate K (equation 5.6) with $1/A$. Additionally we assume that $A \gg 1$. Then the transfer function simplifies to a second order system:

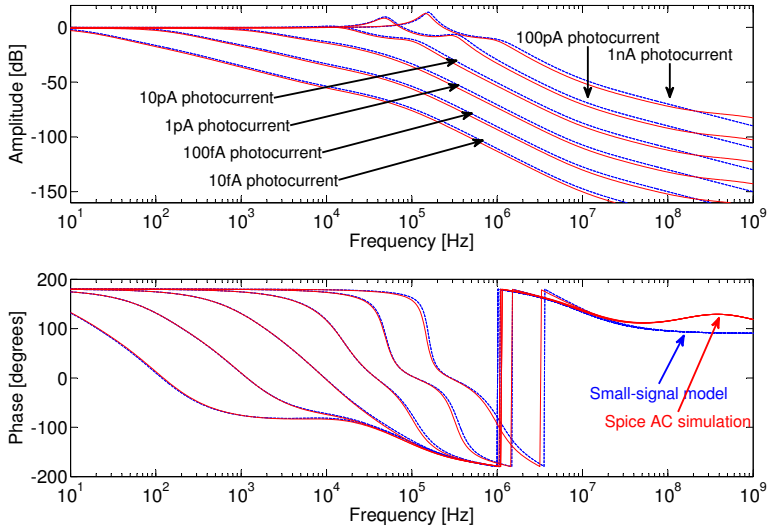
$$\frac{v_s/U_T}{i_s/I_s} = \frac{1}{\kappa_n}\frac{1+s\tau_{up} - s\frac{\tau_u}{A}\frac{i_t}{i_s}}{(1+s\tau_{up})(1+s\tau_{lo}) - s^2\frac{\tau_t^2}{A^2}} \tag{5.12}$$

$$\frac{v_t/U_T}{i_t/I_t} = -\frac{1}{\kappa_p}\frac{1+s(\tau_n + \frac{\tau_b}{A} - \frac{C_u}{Ag_{mfn}}\frac{i_b}{i_t})}{(1+s\tau_{up})(1+s\tau_{lo}) - s^2\frac{\tau_t^2}{A^2}} \tag{5.13}$$

where we used $\tau_{up} = \tau_p + \tau_u/A$ and $\tau_{lo} = \tau_n + \tau_w/A$.



(a) $\frac{v_s/U_T}{i_s/I_s}$, amplitude and phase plot



(b) $\frac{v_t/U_T}{i_t/I_t}$, amplitude and phase plot

Figure 5.2: Comparison of bode plots for small signal model (blue, dashed line) and spice simulation (red, solid line). The photocurrents range from 10fA to 1nA, the bias current is 1nA. The small signal parameters are: $C_u = 144fF, C_b = 13.8fF, C_m = 0.31fF, C_o = 5fF, g_{ma} = 36nS, g_o = 570pS, g_{mfn} = 0.8pS \ldots 80nS, g_{mfp} = 0.3pS \ldots 30nS$.

The denominator for both equations is

$$1 + s(\tau_{up} + \tau_{lo}) + s^2(\tau_{up}\tau_{lo} - \frac{\tau_t^2}{A^2}). \tag{5.14}$$

For a second order system to be stable, all the coefficients of the characteristic polynomial of the denominator have to be positive. Therefore, the condition for stability is

$$\tau_{up}\tau_{lo} - \frac{\tau_t^2}{A^2} > 0 \tag{5.15}$$

$$\tau_n\tau_p + \frac{\tau_w\tau_u}{A^2} + \frac{\tau_n\tau_u}{A} + \frac{\tau_w\tau_p}{A} > \frac{\tau_t^2}{A^2}, \tag{5.16}$$

which is always fulfilled, because

$$\frac{\tau_w\tau_u}{A^2} = \frac{C_u(C_u + C_b)}{A^2 g_{mfn} g_{mfp}} > \frac{\tau_t^2}{A^2} = \frac{C_u^2}{A^2 g_{mfn} g_{mfp}}. \tag{5.17}$$

The roots of the denominator are

$$s_{1,2} = \frac{-(\tau_{up} + \tau_{lo}) \pm \sqrt{(\tau_{up} + \tau_{lo})^2 - 4(\tau_{up}\tau_{lo} - \frac{\tau_t^2}{A^2})}}{2(\tau_{up}\tau_{lo} - \frac{\tau_t^2}{A^2})} \tag{5.18}$$

$$s_{1,2} = \frac{-(\tau_{up} + \tau_{lo}) \pm \sqrt{(\tau_{up} - \tau_{lo})^2 + 4\frac{\tau_t^2}{A^2}}}{2(\tau_{up}\tau_{lo} - \frac{\tau_t^2}{A^2})}. \tag{5.19}$$

The discriminant is always positive, therefore this circuit is not ringing.

Fig. 5.3 compares bode plots for the simplified transfer function (equation 5.12 and 5.13) to the full transfer function (equation 5.8 and 5.9). The curves show a good agreement up to a few kilohertz.

To make the assumption of a sufficiently fast amplifier ($A\tau_o, A\tau_m \ll \tau_w, \tau_u, \tau_n, \tau_p$) valid also for higher frequencies, the bias currents through transistors M_{biasP} and M_{biasN} have to be increased. To always satisfy these conditions without burning unnecessary power in low light conditions, adaptive biasing might be beneficial.

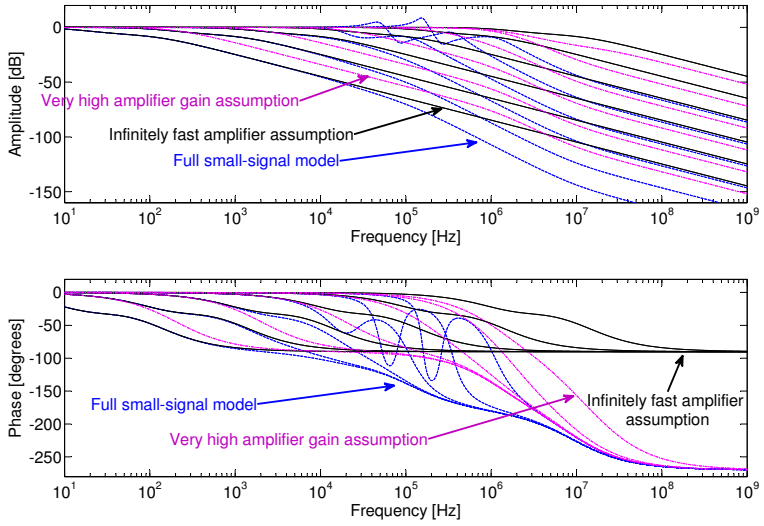
5.1.2 Assumption 2: very high amplifier gain

If the amplifier gain is very high, the nodes V_a and V_w do hardly move, the time constants defined by C_u and C_b will be negligible compared to the time constants defined by the miller capacitance ($\tau_n > \tau_w/A, \tau_p > \tau_u/A,$). Therefore there will be no coupling anymore between V_w and V_a, and the two front-ends will act independently each like a DVS front-end. We can analyze a circuit as shown in Fig. 5.4.

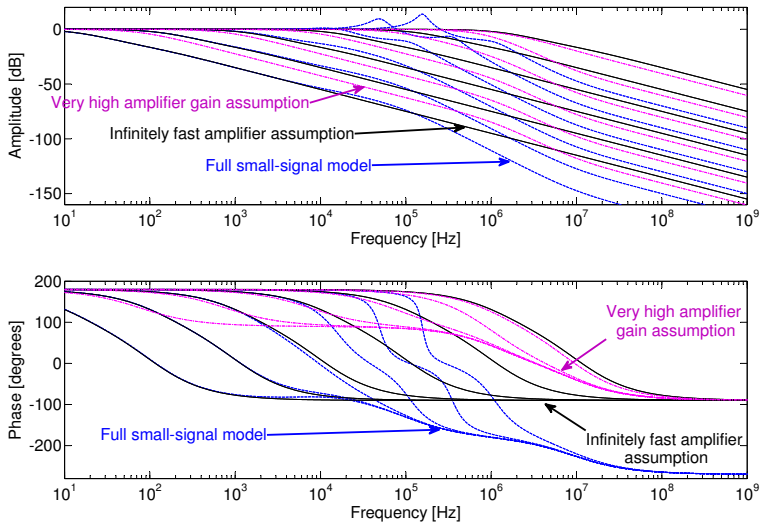
The following equations denote Kirchhoff's current law at the v_{in} and the v_{pr} nodes:

$$i_{\text{in}} = g_{mfb}(v_{\text{pr}} - v_{\text{in}}) + sC_m(v_{\text{pr}} - v_{\text{in}}) \tag{5.20}$$

$$-g_{ma}v_{\text{in}} = sC_m(v_{\text{pr}} - v_{\text{in}}) + sC_o v_{\text{pr}} + g_o v_{\text{pr}} \tag{5.21}$$



(a) $\frac{v_s/U_T}{i_s/I_s}$, amplitude and phase plot



(b) $\frac{v_t/U_T}{i_t/I_t}$, amplitude and phase plot

Figure 5.3: Comparison of bode plots for different small signal models (dasehd blue: full small signal model, black solid: infinitely fast amplifier assumption, magenta dashdot: very high amplifier gain assumption). The photocurrents range from 10fA to 1nA, the bias current is 1nA. The small signal parameters are: $C_u = 144fF, C_b = 13.8fF, C_m = 0.31fF, C_o = 5fF, g_{ma} = 36nS, g_o = 570pS, g_{mfn} = 0.8pS \ldots 80nS, g_{mfp} = 0.3pS \ldots 30nS$.

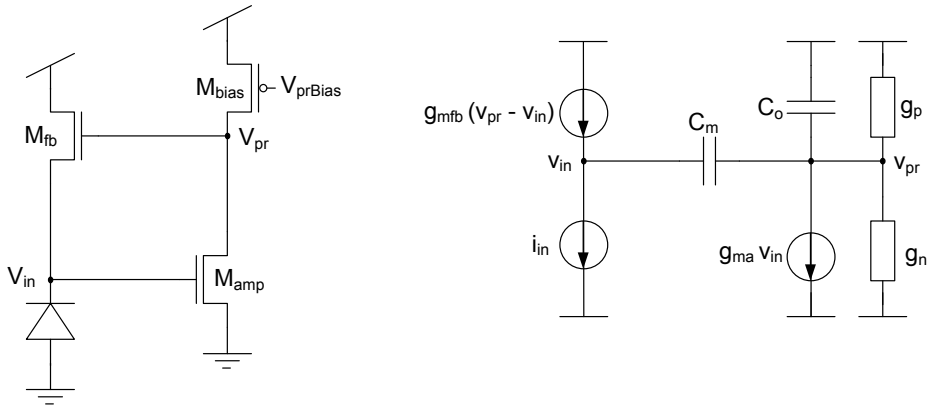


Figure 5.4: DVS front-end and corresponding small-signal model

The output conductance g_o is the sum of the drain-source conductances $g_p + g_n$ of the amplifier transistors. Solving for the transfer function and using the same time constant definitions as before leads to

$$\frac{v_{\mathrm{pr}}/U_T}{i_{\mathrm{in}}/I_{\mathrm{ph}}} = \frac{1}{\kappa_{fb}} \frac{1 - s\tau_m}{(1 + s\tau_n)(1 + \frac{1}{A} + s\tau_o)}. \tag{5.22}$$

Assuming $\tau_n > \tau_o$ (which makes sense because usually $C_o > C_m$ and $I_{\mathrm{ph}} < I_{\mathrm{prBias}}$), the dominant time constant of the system is the Miller capacitance divided by the transconductance of the feedback transistor, which is proportional to the photocurrent I_{ph}. Increasing the open loop gain of the common source amplifier does not decrease the time constant due to the miller effect.

For the color pixel front-end, the following transfer functions will arise:

$$\frac{v_s/U_T}{i_s/I_s} = \frac{1}{\kappa_n} \frac{1 - s\tau_m}{(1 + s\tau_n)(1 + \frac{1}{A} + s\tau_o)} \tag{5.23}$$

$$\frac{v_t/U_T}{i_t/I_t} = -\frac{1}{\kappa_p} \frac{1 - s\tau_m}{(1 + s\tau_p)(1 + \frac{1}{A} + s\tau_o)} \tag{5.24}$$

These transfer functions are stable and not ringing. Fig. 5.3 compares bode plots for this simplification to the full transfer function of equations 5.8 and 5.9. This figure shows that the agreement of this model and the full small signal model is not very well for this set of parameters.

5.1.3 Combination of both assumptions

If we combine both above mentioned assumption of an infinitely fast amplifier with very high gain, we arrive at a simple first order low-pass characteristic.

$$\frac{v_s/U_T}{i_s/I_s} = \frac{1}{\kappa_n} \frac{1}{1 + s\tau_n} \tag{5.25}$$

$$\frac{v_t/U_T}{i_t/I_t} = -\frac{1}{\kappa_p} \frac{1}{1 + s\tau_p} \tag{5.26}$$

The bandwidth is proportional to the photocurrents and because $I_s > I_t$, the bandwidth of the n-type front-end will always be bigger than the bandwidth of the p-type front-end. Additionally, the ratio of the bandwidths will depend on the light color, which can be problematic and lead to a transient, band-pass like response in the output voltage for a step intensity change (which is a step change in both photocurrents without changing their ratio).

5.2 Effects of a Cascode Transistor

To investigate the effects of a cascode transistor in the inverting amplifier, we compare the small signal transfer function of an active logarithmic front-end without (subsection 5.1.2, Figure 5.4) and with cascode (Figure 5.5).

In a single stage common source amplifier, a cascode transistor serves two basic properties. It increases the output impedance (and therefore the open loop gain, see Johns and Martin, page 140 [123]) and reduces the capacitance between input and output node. This capacitance (the gate-drain overlap capacitance of the n-FET transistor) can limit the bandwidth even though it is usually quite small, because the input sees a capacitance which is multiplied by the gain (this is the so-called Miller-Effect, see Johns and Martin, page 154 [123]). The cascode largely nullifies this effect, because the gate-drain overlap capacitance is now not connected to the output anymore.

However, in the DVS circuit, the Miller capacitance is formed by the parallel gate-drain overlap capacitance of the n-FET amplifier transistor M_{amp} and the gate-source capacitance of the feedback transistor M_{fb}. The cascode transistor removes the effect of the M_{amp} gate-drain capacitance, but not of the M_{fb} gate-source capacitance. Therefore the Miller capacitance in the corresponding small signal equivalent circuit is half of the value of the circuit without cascode, assuming that the feedback transistor and the amplifier transistor have equal widths.

In these calculations, we assume that the open loop gain of the common-source amplifier is high enough so that the time constant defined by the parasitic photodiode capacitance can be neglected.

5.2.1 Front-end with cascode

For the analysis, we assume that the amplifier input transistor and the cascode transistor have equal geometries, which leads to equal transconductance and drain-source conductance. Fig. 5.5 shows the circuits and the corresponding small signal model.

The following equations denote Kirchhoffs current law at the v_{in}, v_{cas} and the v_{pr} nodes.

$$i_{\text{in}} = g_{mfb}(v_{\text{pr}} - v_{\text{in}}) + s\frac{C_m}{2}(v_{\text{pr}} - v_{\text{in}}) \tag{5.27}$$

$$g_{ma}v_{\text{in}} + g_n v_{\text{cas}} = -g_{ma}v_{\text{cas}} + g_n(v_{\text{pr}} - v_{\text{cas}}) \tag{5.28}$$

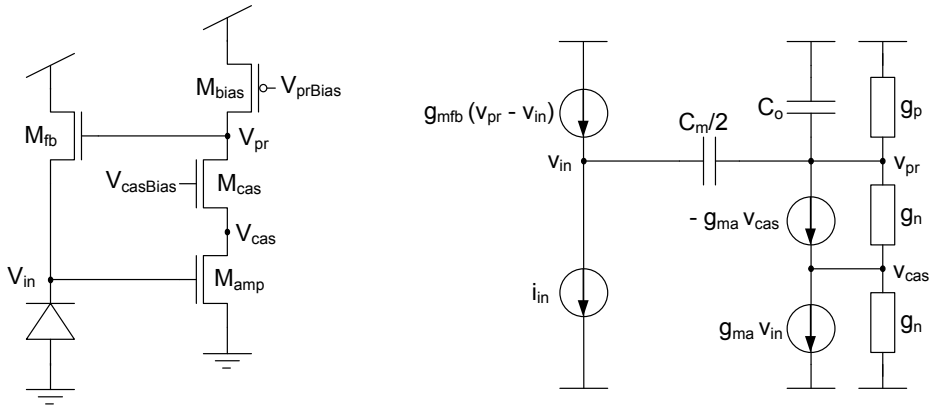


Figure 5.5: DVS front-end with cascode transistor and corresponding small-signal model

$$s\frac{C_m}{2}(v_{\text{in}} - v_{\text{out}}) = -g_{ma}v_{\text{cas}} + g_n(v_{\text{out}} - v_{\text{cas}}) + sC_o v_{\text{pr}} + g_p v_{\text{pr}} \tag{5.29}$$

This leads to the following transfer function:

$$\frac{v_{\text{pr}}/U_T}{i_{\text{in}}/I_{\text{ph}}} = \frac{1}{\kappa_{fb}}\frac{1 - s\tau_m}{(1 + s\frac{\tau_n}{2})(1 + \frac{1}{A} + s\tau_o)}. \tag{5.30}$$

We can see that the cascode speeds up the response by a factor of two (assuming that $\tau_n/2 > \tau_o$) compared to equation 5.24. This is due to the effective removal of the gate-drain overlap capacitance of the n-FET amplifier transistor, which results in halving the Miller capacitance between photodiode and output node. The increased gain of the amplifier does not speed up the response of the system because of the Miller effect.

However, the increased open loop gain of the common source amplifier with cascode can nevertheless speed up the circuit in the case where the parasitic capacitance of the photodiode is not negligible compared to the Miller effect. Due to higher open loop gain the photodiode node will have to move less, the time constant of the parasitic photodiode capacitance is inversely proportional to the amplifier gain A.

Therefore, the effective speedup of the cascode depends on the ratio of the parasitic capacitances and the open loop amplifier gain. If the Miller-effect is limiting the bandwidth (e.g. $AC_m \gg C_{ph}$), the cascode provides a speed-up of two, if the pole caused by the photodiode is dominant (e.q. $AC_m \ll C_{ph}$), the speed-up is equal to the increase in open loop gain, which depends on the ratio of g_n and g_p.

5.3 Noise analysis

In the DVS pixel [66, 39], the pixel bandwidth is dependent on the illumination level for low to intermediate illumination levels. Higher noise power spectral density at

low illumination levels is compensated by lower bandwidth, leading to constant RMS noise voltage levels at the output for all but the lowest intensities. At very low intensities the photocurrent becomes comparable to the dark current and the RMS noise level at the output increases.

The color change pixel is somewhat different, as the first stage of the amplifier A_1 is deliberately used to limit the bandwidth at all but the lowest intensities. This means that the RMS noise voltage at the output decreases with increasing intensity.

For low illumination but still significantly above the dark current, the dominant noise sources are the photon shot noise of the two junctions of the BDJ and the two feedback transistors M_{fbP} and M_{fbN} [124]. Each of these noise sources has a flat power spectral density of $S(f) = \overline{\Delta i^2}/\Delta f = 2qI$ [125], where q is the elementary charge and I is the current through the respective device.

The spectra of the noise sources are shaped by the transfer functions from the noise sources to the output. Using the first order approximations of the front-end according to equations 5.25 and 5.26 and assuming a first order low pass with a time constant τ_A for the amplifier, the transfer functions from the noise sources to the output are as follows.

$$\mathrm{TF_{fNn}}(s) = \frac{v_{\mathrm{coldiff}}}{i_{\mathrm{mfNn}}} = -\frac{\kappa A_c}{g_{mfn}(1+s\tau_A)(1+s\tau_n)} \tag{5.31}$$

$$\mathrm{TF_{fPn}}(s) = \frac{v_{\mathrm{coldiff}}}{i_{\mathrm{bn}}} = \frac{\kappa A_c}{g_{mfn}(1+s\tau_A)(1+s\tau_n)} \tag{5.32}$$

$$\mathrm{TF_{bn}}(s) = \frac{v_{\mathrm{coldiff}}}{i_{\mathrm{mfPn}}} = \frac{\kappa A_c}{g_{mfp}(1+s\tau_A)(1+s\tau_p)} \tag{5.33}$$

$$\mathrm{TF_{tn}}(s) = \frac{v_{\mathrm{coldiff}}}{i_{\mathrm{tn}}} = \frac{\kappa A_c(1/g_{mfp} - 1/g_{mfn})}{(1+s\tau_A)(1+s\tau_n)(1+s\tau_p)} \tag{5.34}$$

Here A_c stands for the gain of the capacitive amplifier, i_{mfNn} is the noise current source from transistor M_{fbN}, i_{mfPn} is the noise current source from M_{fbP}, i_{bn} is the noise current source from the bottom junction and i_{tn} is the noise current source from the top junction.

To calculate the total noise power spectral density at V_{coldiff}, we have to sum the different noise current source contributions.

$$S_o(f) = |\mathrm{TF_{fNn}}|^2\frac{\overline{\Delta i^2_{\mathrm{mfNn}}}}{\Delta f} + |\mathrm{TF_{fPn}}|^2\frac{\overline{\Delta i^2_{\mathrm{mfPn}}}}{\Delta f} + |\mathrm{TF_{bn}}|^2\frac{\overline{\Delta i^2_{\mathrm{bn}}}}{\Delta f} + |\mathrm{TF_{tn}}|^2\frac{\overline{\Delta i^2_{\mathrm{tn}}}}{\Delta f} \tag{5.35}$$

$$S_o(f) = \frac{2q\kappa^2A_c^2}{1+\frac{f^2}{f_A^2}}\left(\frac{2I_B+I_T}{g_{mfn}^2(1+\frac{f^2}{f_n^2})} + \frac{I_T}{g_{mfp}^2(1+\frac{f^2}{f_p^2})} + \frac{I_T(g_{mfn}-g_{mfp})^2}{g_{mfp}^2g_{mfn}^2(1+\frac{f^2}{f_n^2})(1+\frac{f^2}{f_p^2})}\right) \tag{5.36}$$

To simplify the further analysis, we assume that $I_B = I_T = I_{ph}$ and $g_{mfn} = 2g_{mfp}$. Additionally we assume that $f_n \approx f_p = f_r$, even though in reality $f_n \approx 2f_p$ for $I_B = I_T$.

$$S_o(f) \approx \frac{2qI_{ph}\kappa^2A_c^2}{g_{mfp}^2\left(1+\frac{f^2}{f_A^2}\right)}\left(\frac{1}{1+\frac{f^2}{f_r^2}} + \frac{1}{4\left(1+\frac{f^2}{f_r^2}\right)^2}\right) \tag{5.37}$$

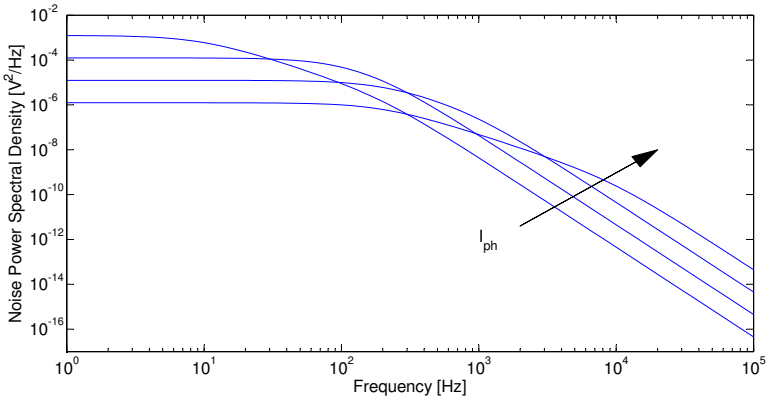


Figure 5.6: Noise power spectral density.f_A is assumed to be $200Hz$, C_m is $1fF$, A_c is 100.

$$S_o(f) \approx \frac{2qU_T^2A_c^2}{I_{ph}\left(1+\frac{f^2}{f_A^2}\right)\left(1+\frac{f^2}{f_r^2}\right)}\left(1+\frac{1}{4\left(1+\frac{f^2}{f_r^2}\right)}\right) \tag{5.38}$$

The noise power spectral density is plotted in Fig. 5.6 for different photocurrents ranging from one femtoampere to one picoampere.

The total noise power at the output is the integral of the power spectral density.

$$\overline{\Delta V_{\text{coldiff}}^2} = \int_0^\infty S_o(f)\,\mathrm{d}f \tag{5.39}$$

To be able to find a closed form of the integral of $S_o(f)$, we have to simplify the last term of $S_o(f)$. For the front-end bandwidth, we use $f_r = \kappa I_{ph}/(2\pi U_T C_m)$.

$$S_o(f) \approx \frac{2qU_T^2A_c^2}{I_{ph}(1+\frac{f^2}{f_A^2})(1+\frac{f^2}{f_r^2})}\frac{5}{4} \tag{5.40}$$

$$\overline{\Delta V_{\text{coldiff}}^2} = \int_0^\infty S_o(f)\,\mathrm{d}f = \frac{\frac{5}{2}\pi A_c^2U_T^2qf_A}{I_{ph}(1+\frac{f_A}{f_r})} \tag{5.41}$$

$$\overline{\Delta V_{\text{coldiff}}^2} = \frac{\frac{5}{4}A_c^2U_T^2q\omega_A}{I_{ph}+\frac{\omega_AU_TC_m}{\kappa}}, \tag{5.42}$$

Now we can calculate a closed form for the RMS noise voltage

$$v_{\text{RMS}} = \sqrt{\overline{\Delta V_{\text{coldiff}}^2}} \tag{5.43}$$

$$= \frac{A_cU_T}{2}\sqrt{\frac{5q\omega_A}{I_{ph}+\frac{\omega_AU_TC_m}{\kappa}}} \tag{5.44}$$

The RMS noise voltage versus photocurrent is plotted in Fig. 5.7 for $f_A = 200$Hz, $A_c = 100$ and $C_m = 0.6fF$.

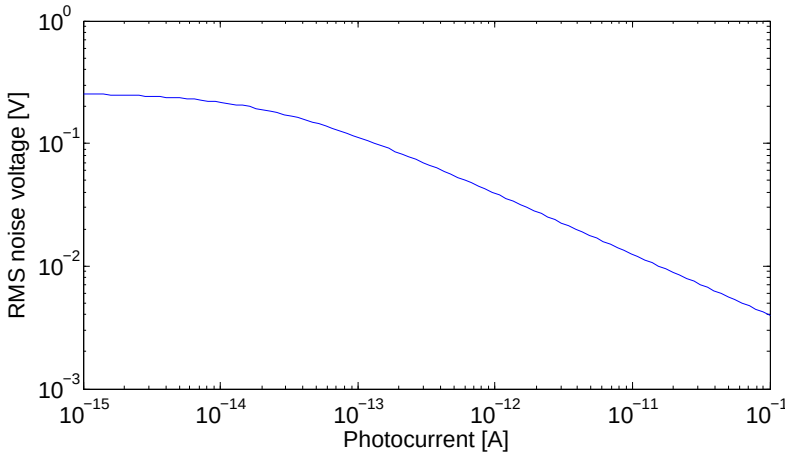


Figure 5.7: Photocurrent versus output RMS noise voltage

In our event-based pixels, the noise in the differentiator voltage will finally manifest itself in the jitter of the timing of the events. We expect the latency to be roughly constant for changing intensity, as long as the bandwidth is limited by the amplifier. But the jitter will be reduced with increasing light intensity.

5.4 Discussion

The small signal analysis provided in this chapter leads to some interesting results.

- To minimize the coupling from V_w to V_a through C_u, it is important to make the amplifier gain $A = g_{ma}/g_o$ as big as possible. Increasing the amplifier gain is achieved by making the transistors M_{ampN}, M_{ampP}, M_{prBiasP} and M_{prBiasN} as long as possible.
- If the bandwidth of the front-end is limited by the Miller capacitance, it can only be increased by a factor of two with a cascode transistors.
- As the bandwidth of the two front-ends is different due to the different currents, and the ratio of the bandwidth changes with changing light color, it is beneficial to limit the bandwidth with the differentiator.
- If the bandwidth is limited with the differentiator, the RMS noise voltage in the output will decrease with increasing light intensity. This will decrease jitter for higher illumination and possibly allow lower thresholds. But the lower bound on the event thresholds is not only dependent on RMS noise but also on the mismatch across the array. The thresholds must be set in a way that even in reset, outliers do not hold onto bus all the time.

Chapter 6

The combined cDVS color-change and log-intensity change pixel

This chapter describes the combination of the LogSum color change pixel presented in chapter 4 and the DVS [66] log intensity change pixel. It provides measurements from an implementation in UMC 6M1P 180nm technology.

6.1 The Pixel circuit

The cDVS pixel combines circuits that detect both changes in log intensity and changes in mean wavelength. A single buried double junction provides input to both color and monochrome pathways (Fig. 6.1). These pathways have separate outputs: The color pathway outputs binary events signaling increases or decreases of wavelength and the monochrome pathway outputs binary events signaling increases or decreases of brightness.

6.1.1 cDVS color change pathway

The color change pathway (Fig. 6.2) computes changes in the I_S/I_T ratio by amplifying the change in the difference of the logarithms of I_S and I_T. The circuit uses the common BDJ, two logarithmic current to voltage photoreceptors (one of which is shared with the log-intensity change pathway), a two-stage summing amplifier, "redder" and "bluer" comparators, and a reset and refractory circuit. Whenever V_{coldiff} crosses threshold, the summing amplifier is reset by closing and then opening the switches M_{r1} and M_{r2}, thereby memorizing the last difference.

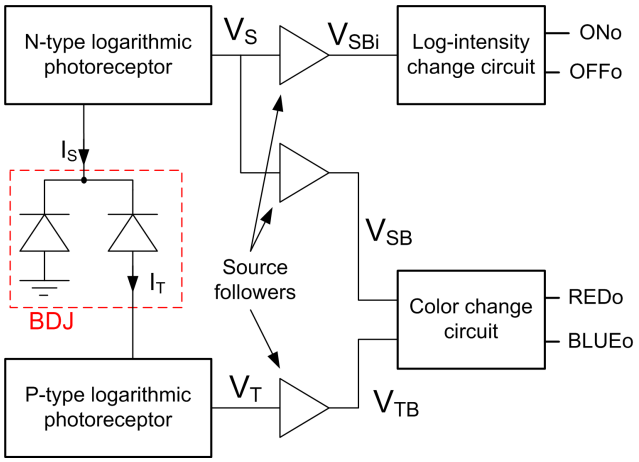


Figure 6.1: Block diagram of cDVS combined color change and log-intensity change detection pixel.

Modifications compared to ColTmpDiff LogSum pixel

The color change circuit is described in section 4.1.1, but incorporates changes to the reset circuit which were intended to reduce the effects of charge injection, as well an the additional pFET cascode transistor M_{cas}.

This transistor is to speed up the response of the p-type photoreceptor relative to the n-type photoreceptor at low photo-currents. It is only used in the p-type photoreceptor because the bandwidth of both photoreceptors is proportional to the photocurrent (see chapter 5 or [66]), but the current in the n-type receptor is always bigger because $I_S > I_T$.

The source of M_{amp} of A_1 is connected to shifted source voltage V_{SSN} which allows biasing with low currents, so we can set the low pass cut-off frequency of A_1 to low frequencies [126].

As reasoned in section 4.1.1, the assumption that the response to intensity change is minimal when the input capacitances to the summing amplifier are equal (and thus $A_S = A_T$) seems justified because the transistors M_{N1} and M_{N4} (respectively M_{P1} and M_{P4}) have the same geometry and the same gate-bulk voltage. Also, the ratio of the currents through these transistors $I_{M_{N1}}/I_{M_{N4}}$ and $I_{M_{P1}}/I_{M_{P4}}$ are similar, because the photocurrents are of the same order of magnitude and the bias currents through M_{N4} and M_{P4} are the same. This would mean that the slopes of V_{TB} and V_{SB} versus photocurrent should be the same.

However, T-SPICE DC current sweep simulations (Fig. 6.3, using the manufacturer supplied models, BSIM3V3.2) showed a difference in the slope of V_{TB} and V_{SB} of 14%, independent of the photoreceptor bias current (bias currents through M_{P3} and M_{N3} are the same) or the source follower bias current (not shown). That the slopes are quite different is nicely illustrated in Fig. 6.3(b), where the sum of the voltages V_{TB} and V_{SB} is plotted versus photocurrent. According to the reasoning in

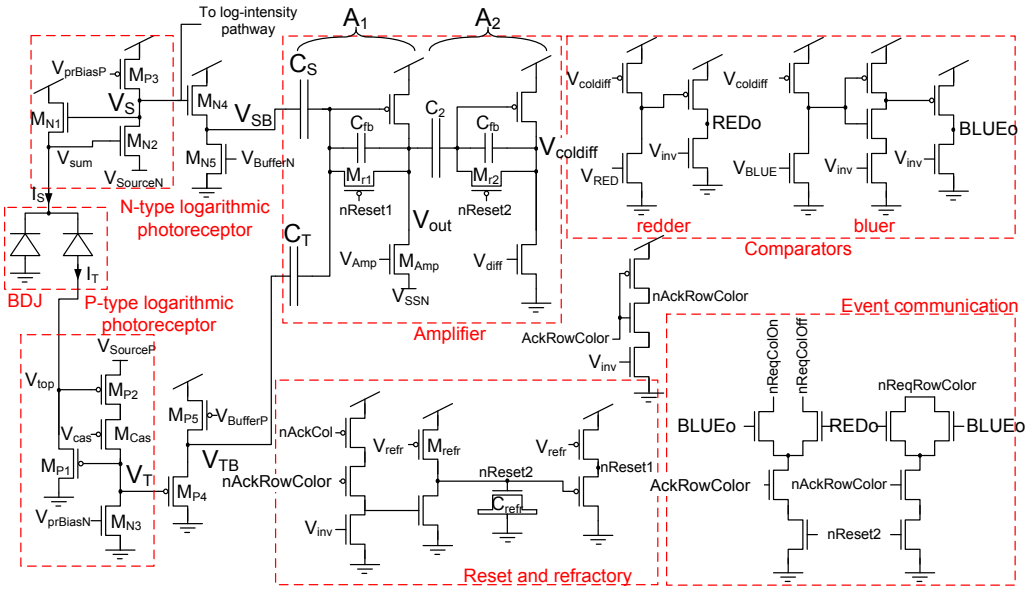
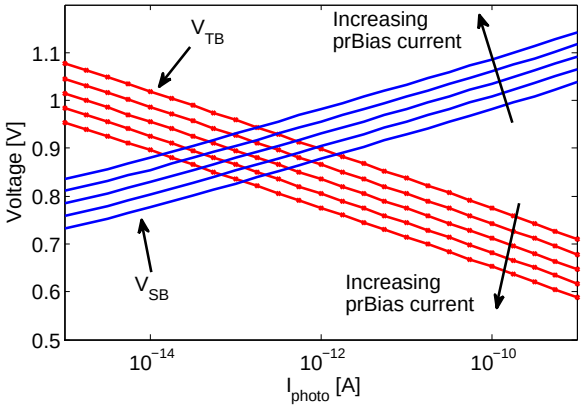


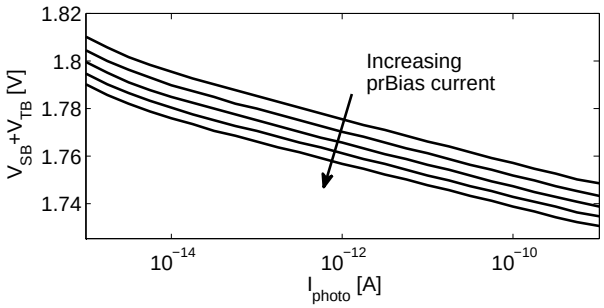
Figure 6.2: Color-change pathway of the cDVS pixel. The column request signals *nReqColOn* and *nReqColOff* are shared with the intensity change detection pathway, the row request signals are separate.

the last paragraph, the sum of V_{TB} and V_{SB} should be independent of photocurrent for a constant ratio I_S/I_T. But the Spice simulation suggested that C_S should be made 14% larger than C_T for the smallest possible response to intensity changes, so we designed this ratio into the pixel capacitor layout. Our measurements in (Fig. 6.9 in Sec. 6.3) show that we should have trusted basic theory more than device simulation.

Event communication and amplifier reset occur when the signals *BLUEo* and *REDo* pull the active-low row request *nReqRowColor* to ground (Fig. 6.4). When the pixel receives the active-high row acknowledge *AckRowColor*, it activates either the *nReqColOff* or *nReqColOn* column request, depending on *BLUEo* and *REDo*. As soon as both row and column acknowledge signals are active, *nReset2* is pulled to ground, which closes the reset switch M_{r2}. The reset "refractory" period (the slew rate of *nReset2* returning to V_{dd}) is controlled by the bias V_{refr}. The reset switch of A_1, M_{r1}, is controlled by *nReset1*, which is generated with a p-type source follower from *nReset2*. The voltage on *nReset1* is higher than the voltage on *nReset2*, which ensures that A_2 is held longer in reset than A_1 so that charge injection from opening M_{r1} can settle. Posch et al. [119] generate *nReset1* with two inverters in series from *nReset2*. Here we use a different implementation with the intention to save area (two transistors vs. four transistors) and avoid the large power consumption caused by the slowly switching inverters. As we will see, our implementation with no degree of freedom and similarly slow slopes for both reset



(a) cDVS photoreceptor DC responses.



(b) Sum of V_{SB} and V_{TB}

Figure 6.3: Simluation of cDVS photoreceptor DC responses. For these simulations it is assumed that $I_B = I_T = I_{\mathrm{photo}}$, the bias currents I_{prBias} through transistors M_{P3} and M_{N3} are the same. The plots illustrate that the slopes of the DC response do not depend on the bias current.

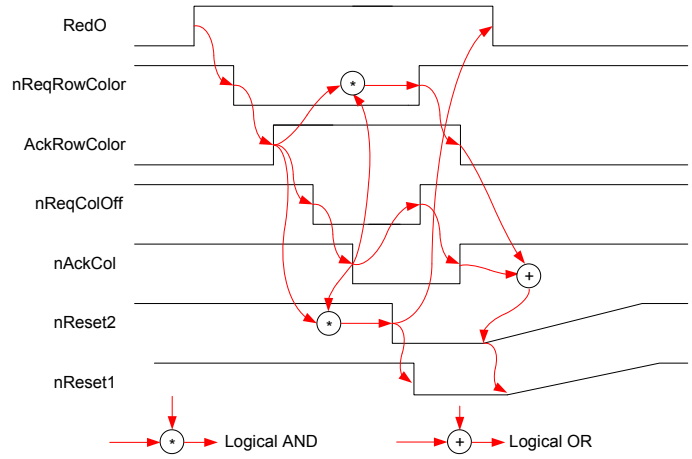


Figure 6.4: Timeline of the signals involved in event communication for a redder event.

signals is a bad idea that leads to an unavoidable large offset voltage being stored at the output of A_2.

In an array of pixels, the row and column requests would be connected to arbiters [76], but this test pixel acknowledges itself through external logic circuits that are not shown here.

6.1.2 cDVS log-intensity change pathway

Because the part of the cDVS pixel that detects log-intensity changes is very similar to the DVS (section 1.5, [66]), here I only present a short summary.

The log-intensity change detection circuit (Fig. 6.5) responds asynchronously to log intensity changes. Each emitted event represents a quantized change in log intensity. Its input is the summed BDJ current I_S. We arbitrarily define this photocurrent as the measure of "intensity". It uses the cDVS pixel n-type active logarithmic photoreceptor whose output V_S is buffered with a source follower to produce V_{SBi}. Each log-intensity change event causes the log intensity to be memorized across capacitor C_1. The log intensity change is amplified by a switched capacitor amplifier. Comparators detect changes in the log intensity value, resulting in *OFFo* or *ONo* events. The BDJ and the logarithmic photoreceptor are common to the intensity and the color pathway. The AER communication of events is identical to the color pathway.

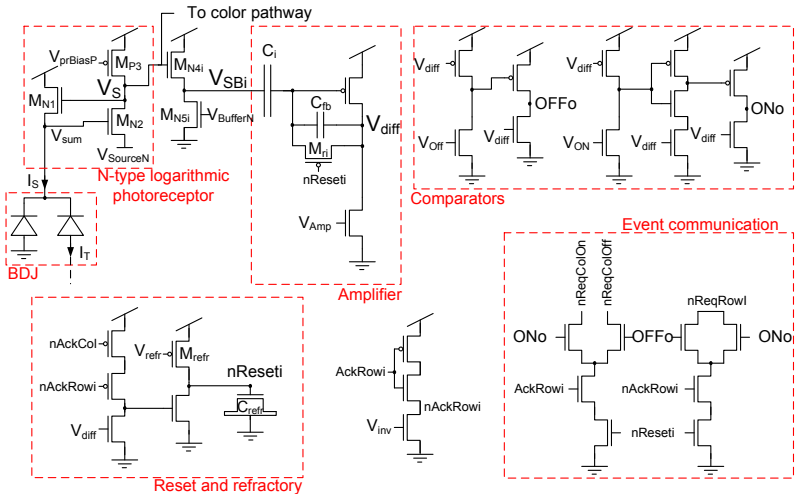


Figure 6.5: Log-intensity change pathway for the cDVS pixel. The column request signals *nReqColOn* and *nReqColOff* are shared with the color change detection pathway, the row request signals are separate.

6.2 cDVSTest10 chip

To test this pixel design along with the new word serial AER circuits presented in chapter 7, a chip was designed and fabricated in the "UMC L180 Mixed-Mode/RF" 180nm 6-metal 1-poly process with metal-insulator-metal (MIM) capacitors [99]. The test chip incorporated a 32×32 array of cDVS pixels, four 32×32 array of new DVS pixel designs, a new fully configurable bias generator, a spectral sensitivity measurement circuit as described in section 2.4.2 as well as a separate cDVS test pixel independent of the arrays.

Unfortunately, we forgot to connect the reset signal of the AER circuits to a pad, which made the AER circuits and therefore the pixel array non-functional. However, we could still measure the performance of the cDVS test pixel.

Table 6.1 lists pixel specifications and Fig 6.6 shows the pixel layout. The pixel is drawn for use in a quad-mirrored array configuration [39]. The pixel size of $29 \times 29 \mu m^2$ was imposed by the AER peripheral circuits. The 330fF of MIM capacitance is stacked above the circuits.

6.2.1 Bias generator

The cDVSTest10 chip includes the latest generation of programmable bias-current generators [118, 127, 77, 126], designed by Tobi Delbruck.

This newest implementation is fully configurable and improves several things over the latest published implementation [126]. The shifted sources for V_{SSN}, V_{sourceN} and V_{sourceP} have a more flexible voltage generation circuit than before.

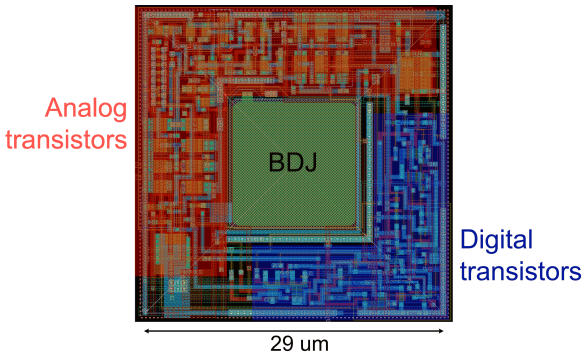


Figure 6.6: Layout of the cDVS pixel.

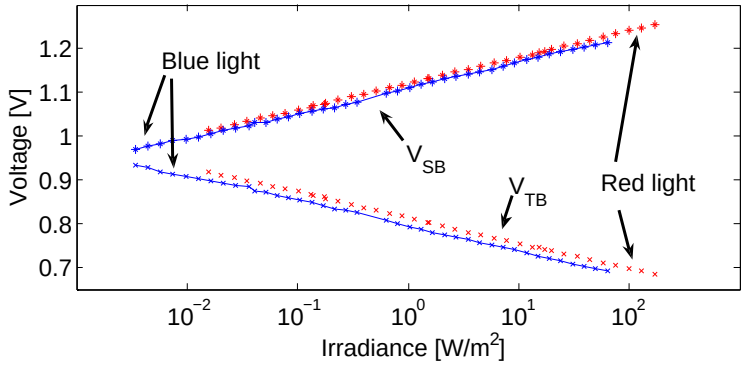
Process	UMC 180 nm, N-Well, 6 Metal, 1 Poly, MIM-caps
Pixel size	$29\mu\text{m} \times 29\mu\text{m}$
Photodiode Area	$144\mu\text{m}^2$
Fill factor	17%
Number of elements per pixel	82 transistors, 7 capacitors 2 MOS capacitors
Designed Amplifier Gain $\frac{V_{\text{diff}}}{V_{SBi}}, \frac{V_{\text{coldiff}}}{V_{SB}}, \frac{V_{\text{coldiff}}}{V_{TB}}$	27 dB, 46 dB, 47 dB
Power supply	1.8V

Table 6.1: cDVS test pixel specifications

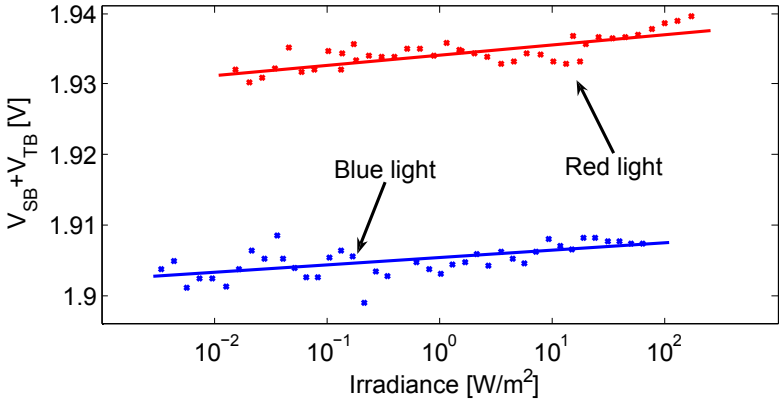
6.3 cDVSTest10 Testpixel Measurements

Figure 6.7(a) shows the DC responses of the voltages V_{SB} and V_{TB} versus intensity for red and blue light. The responses are logarithmic for more than 4 decades spanning from 100W/m^2 chip illumination down to 4mW/m^2. The lowest illumination level corresponds approximately to about 1 lux of white light, which corresponds to scene illumination of about 50 lux when using a fast lens [128]. Figure 6.7(b) shows $V_{\text{SB}} + V_{\text{TB}}$ versus intensity for two different light colors. Due to the current dependence of κ, $V_{\text{SB}} + V_{\text{TB}}$ changes by about 5mV over 4 decades of light intensity with constant color. A color change from red to blue at constant intensity changes $V_{\text{SB}} + V_{\text{TB}}$ by roughly 30mV. The change in $V_{\text{SB}} + V_{\text{TB}}$ with wavelength (or, non-ideally, with intensity) is amplified by the capacitive summing circuit to generate color change events.

As mentioned in Sec. 6.1.1 just after Eq. 4.6, SPICE simulations suggested



(a) cDVS photoreceptor responses



(b) Sum of V_{SB} and V_{TB}

Figure 6.7: cDVS DC response of the photoreceptors.

that the slopes of V_{TB} and V_{SB} with intensity would differ by 14%. However, the measurement (Fig. 6.7(a)) shows that these slopes differ by only two percent, closely in accordance with fundamental reasoning about transistor behavior. Because we used the value for C_S/C_T suggested by the SPICE simulations, the cDVS response to intensity change in the color change pathway is 7 times larger than it would have been had we used a ratio of one for C_S/C_T. These measurements suggest that the current-level dependence of κ may be poorly modeled in SPICE. However we did not further investigate to find the cause for this discrepancy between simulation and measurements.

Figure 6.8 shows the measured noise spectral density over three decades of illumination. The low frequency noise is similar for the higher-illumination ND1 and ND2 curves, suggesting that at these illumination levels, other noise sources besides shot noise of the junctions and the feedback transistors dominate. For the lowest illumination level ND3 curve, the Eq. 5.38 noise sources dominate and the noise is inversely proportional to intensity. The roll-off of the ND3 curve becomes

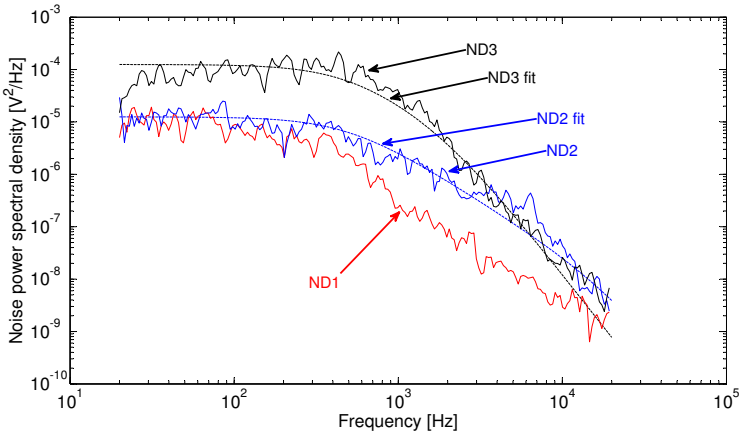
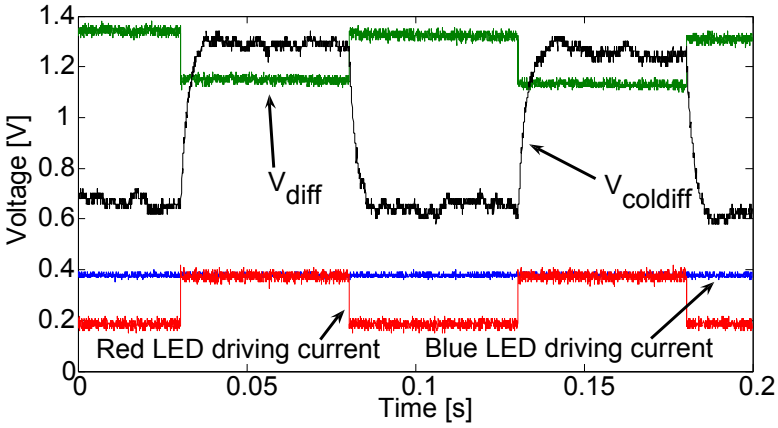


Figure 6.8: Noise power spectral density of V_{coldiff} for different illumination levels. NDx means that a neutral density filter of density x in decades was used to attenuate the constant light source.

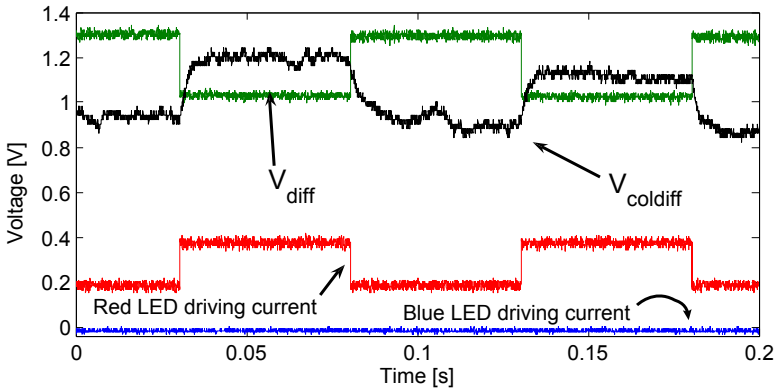
steeper above 2kHz because the cutoff frequency of the photoreceptor is similar to that of A_1.

The theoretical fit (using equation 5.40) of the ND2 curve shows good agreement with the measured data using photocurrent values of 800fA, an amplifier corner frequency of 500Hz and a front-end corner frequency of 20kHz; the total V_{coldiff} RMS noise is 110mV, corresponding to an input-referred contrast RMS noise of 4.5%. The theoretical fit for the ND3 curve uses a photocurrent value of 80fA, an amplifier corner frequency of 500Hz and a front-end corner frequency of 2kHz, in nice agreement to a factor 10 attenuation by the neutral density filter. Lower intensities produce more noise in the passband of A_1, making event detection less reliable. The high level of input-referred noise here is due to the high bandwidth of A_1 used in this measurement.

The scope traces of V_{diff} and V_{coldiff} in Fig. 6.9 illustrate that the color pathway has a larger response to color change than to intensity change. V_{coldiff} is the output of the color pathway, V_{diff} the output of the log intensity change pathway. It measures I_{S} and can be used to see how much the intensity changes. In Fig. 6.9(a) we mixed a constant blue LED and a square-wave modulated red LED to produce a color change, along with an intensity change. This input produced a response amplitude of 200mV in V_{diff} and a much larger response of 750mV in V_{coldiff}. In Fig. 6.9(b), we turned off the blue LED and only modulated the red LED, to change only the intensity without changing the color. In Fig. 6.9(b), the LED driving current changes by a factor of two, which causes a one-octave change of intensity and a 300mV response of V_{diff}, which is larger than the previous case, so the intensity changes more. But despite this larger change of intensity, the response of V_{coldiff} is only 250mV, which is smaller than the response to color change, as is desired. But the change is larger than could have been achieved from the the finite



(a) Response to color change



(b) Response to intensity change of one octave.

Figure 6.9: Response of cDVS pixel to square wave input with thresholds set so that no events are produced.

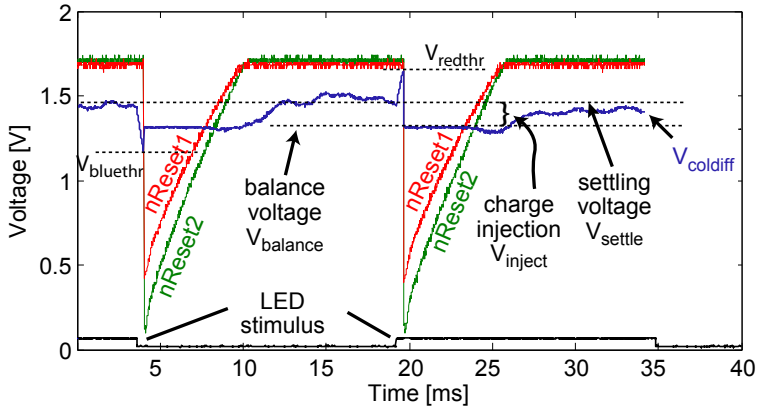


Figure 6.10: cDVS red and blue color change events in response to step changes of color, showing internal signals, threshold levels, and charge injection. No averaging is applied.

slope in $V_{\mathrm{SB}}+V_{\mathrm{TB}}$ in Fig. 6.7(b), because our ratio between A_T and A_S is incorrect, as pointed out earlier. If we had built the pixel with $A_S = A_T = 200$, then this octave step in intensity, which produces less than 0.4mV change in $V_{\mathrm{SB}} + V_{\mathrm{TB}}$, would have resulted in a V_{coldiff} response of only 80mV rather than 250mV.

In the next set of measurements, we adjusted the thresholds so that the pixel could generate color change events. Figure 6.10 shows a close-up on two events generated in response to step color changes. The falling edge of the LED stimulus causes V_{coldiff} to fall to the bluer threshold V_{bluethr} and the rising edge causes it to rise to the redder threshold V_{redthr}. In each case the event causes $nReset2$ and $nReset1$ to go low, which cause V_{coldiff} to reset to its balance voltage V_{balance}. After $nReset2$ finishes slewing high, there is a large charge injection offset V_{inject} of about 100mV in V_{coldiff}, which settles at V_{settle}.

This large positive charge injection effect V_{inject} results because the voltage difference between *nReset1* and *nReset2* cannot be made large enough to keep A_2 in reset until A_1 has settled. Charge from the channel of M_{r1} is injected to the floating node of A_1 until *nReset1* is at V_{dd}, but Fig. 6.10 shows that *nReset1* and *nReset2* reach V_{dd} nearly simultaneously. As a result, the negative charge injection effect at the output of A_1 is amplified and inverted by A_2. The reset circuit described in subsection 6.1.1 is not effective. SPICE simulations of the pixel before (and after) fabrication do not show this large injection effect, because in the simulations, the charge injected onto the input of A_2 from V_{out} and from opening M_{r2} compensate. An easy fix to this problem is to split the generation of the two reset signals, as described in Sec. 6.4 or to use a circuit similar to the one used by Posch et al. [119].

The next measurement shows the events generated in response to continuous color change. For this measurement the LED's were driven with slow 180° out-of-

phase triangular currents (Fig. 6.11), so that the intensity stayed approximately constant. The measurement shows 11 to 12 events for a red to blue transition from 625nm to 450nm wavelength (Fig. 6.11), so each event represents a wavelength change of 15nm. How do we know that these events result from changes in I_S/I_T and not from the undesired responses to intensity change? During the stimulation period, there were less than 3 ON or OFF events from the intensity change pathway produced by changes in I_S. Since each ON or OFF event signals a 15% change of I_S, the total change of intensity is a factor of $1.15^3 = 1.5$, or less than one octave. From Fig. 6.9(b), we can see that a one octave change of intensity produces an output in V_{coldiff} of 200mV. Therefore the intensity change would only account for one of all the color change events. However, Fig. 6.11 again illustrates the problem of charge injection. After each event, V_{coldiff} balances at V_{balance} and then finally settles at a noisy V_{settle} level, quite close to V_{dd}. (The noise variance in V_{settle} is indicated by the shaded region.) As a result, V_{RED} must be set so close to ground that the redder comparator operates much more slowly than does the bluer comparator. That is part of the reason why there is more jitter in the generation of REDDER events. The other reason becomes clear on close inspection of the adjoining BLUER and REDDER events marked with the "*". It can be observed that at the moment of threshold crossing, the slope of V_{coldiff} is much larger in magnitude for the BLUER event than for the REDDER event. It is as if in the redder direction V_{coldiff} leaks away. The reason for this leaking is that in this direction, V_{coldiff} acts as the source of M_{r2} and turns on M_{r2}, which reduces the positive excursion of V_{coldiff}. By contrast, in the bluer direction V_{coldiff} becomes the drain of M_{r2} and can only turn it on to its saturation current level. As a result of this asymmetry in the high-pass filtering effect of the switch, to obtain the same number of REDDER and BLUER events, we must set V_{redthr} closer to V_{settle} than V_{bluethr} is to V_{settle}. Because V_{coldiff} crosses V_{redthr} with a smaller slope than it crosses V_{bluethr}, there is more jitter in the timing of the REDDER events. The jitter in the REDDER events becomes smaller as we increase the stimulus frequency.

We also measured latency of color change responses. Using our nominal bias values, we observed that color change events were emitted within 2ms of a step change of color. The latency was only weakly dependent on intensity over 3 decades of illumination, because the bandwidth is limited by A_1.

6.4 cDVS pixel discussion

The measurements in the previous sections show that the color pathway provides the desired functionality of detecting color change. Our measurements in Fig. 6.11 used thresholds equivalent to a change of 10% in I_S/I_T per color change event, corresponding to 15nm changes in average wavelength. By reducing the thresholds, we can make the pixel somewhat more sensitive to color change, but we are limited by several nonidealities. The undesired response to intensity changes, the large charge injection, and the high noise level mean that several circuit improvements

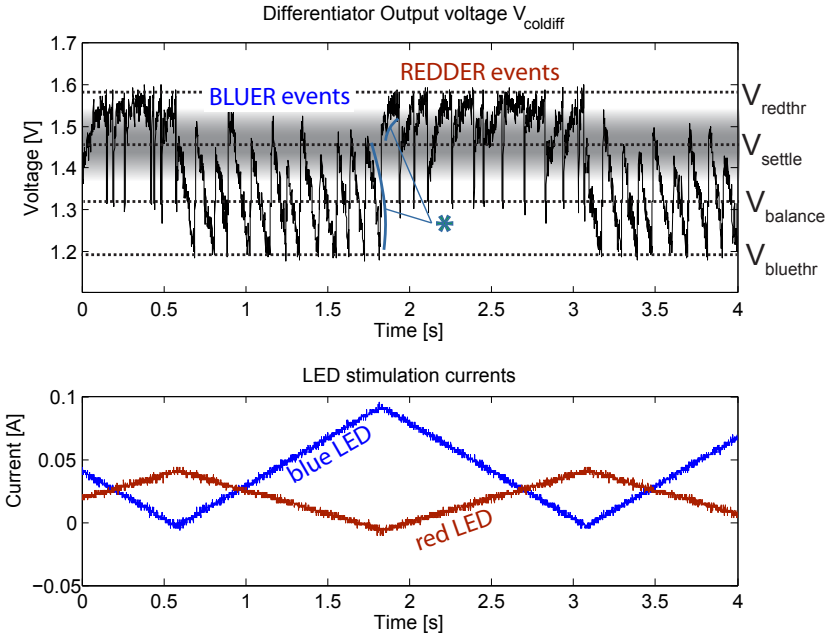


Figure 6.11: cDVS color events in response to color sweeps.

are needed to achieve acceptable real world performance:

1. A correct balance between C_S and C_T will reduce the residual response to intensity changes in the color pathway. Obtaining a correct balance requires careful attention to parasitic capacitances to surrounding structures, such as metal layers.

2. A two-stage capacitive amplifier might still provide acceptable performance if we clearly separated the resets of A_1 and A_2 to that they could definitely be made non-overlapping. This change is easily made by duplicating the reset-and-slew circuit with separate biases for controlling the refractory period of A_1 and A_2 as in the sDVS pixel [80] or using the scheme proposed by Posch et al. [119]. Considering the measurement results of cdVSTest10, we concluded that a two-stage amplifier is not necessary. Using a single stage amplifier simplifies the design and can possibly make the pixel smaller. For example using a single stage amplifier with a gain of $A = 20$ would result in a swing at V_{coldiff} of 600mV in response to a red to blue color change. If we make the assumption that due to mismatch in the comparators and RMS noise, the thresholds need to be set 50mV away from the reset level, then the red to blue transition results in 12 events, which would result in a color change threshold similar to the current implementation.

3. Due to the relatively small signal, the color change pathway benefits from more low pass filtering than does the log intensity change pathway. To be

able to separately control the bandwidths of the color and intensity change pathways even if both pathways use only a single stage amplifier as suggested before, they should have separate V_{diffbias}.

In Chapter 8, we present design and measurement results of a 32×32 pixel array incorporating the changes proposed here. It uses a single stage amplifier with a gain of 27dB for the color change detection.

Chapter 7

The new burst mode word serial AER interface

7.1 Motivation

In the DVS128 vision sensor, the communication bandwidth is limited to around one mega-events per second, which is only about 60 spikes per pixel per second. If we increase the sensor resolution without changing the communication circuits, the per-pixel bandwidth would decrease even more. More efficient AER communication circuits are therefore needed for higher resolution sensors.

7.2 AER scheme used in the DVS128

The AER circuits used in the DVS128 are based on Boahen's earlier work [76], except that the arbiters work in a non-greedy fashion [31].

Here we give only a very short overview of the DSV128 AER circuits, for more detailed descriptions refer to [39, 76]. Fig. 7.1 shows a block diagram of the DV128 AER communication circuits.

The asynchronous communication of an event to a receiver chip works by three sequential four phase handshaking cycles.

1. Row Handshake: A pixel that crossed threshold activates the row request *RR*. When the row arbiter selects the corresponding row, the row address is encoded and the row acknowledge *RA* is activated.

2. Column Handshake: If a row receives the row acknowledge *RA*, all active pixels (pixels that crossed either ON- or OFF-threshold) activate their column request (either *CRon* or *CRoff*, depending on the polarity of the event). The column arbiter will select one column and the corresponding column address is encoded. The column acknowledge *CA* is activated. If a pixel receives both *RA* and *CA*, it is reset.

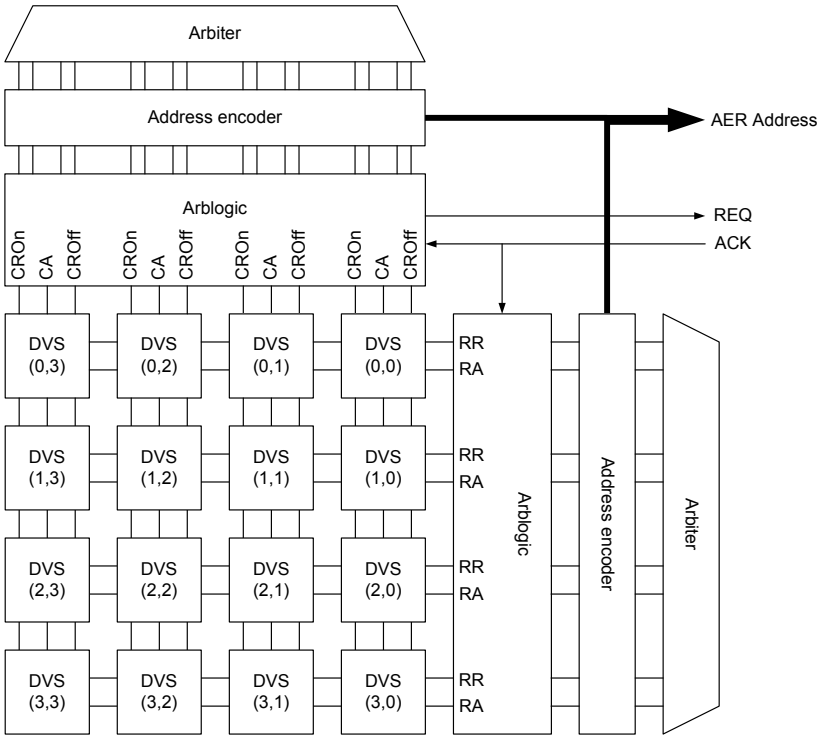


Figure 7.1: DVS128 AER block diagram, reduced array size

3. Chip Handshake: Selection of an active column activates the chip request *REQ*. The row and column address together encode unambiguously the pixel location. As soon as the off-chip receiver activates the chip acknowledge *ACK*, the row and column arblogic circuits are cleared. If there are more active pixels in the same row, the row arbiter will stay at this row and the column arbiter will select a different column. If there are no more active pixels in this row, the row arbiter will select a different row.

The address of the pixel is encoded by a 15-bit data word, 7 bits for the row address, 7 bits for the column address and one bit for the polarity.

The main bandwidth limitation in the DVS128 arises from the time needed to reset the differentiator in the pixel circuit. Even while the differentiator is being reset, the pixel can still be above one of the thresholds during a considerable amount of time (in the order of a microsecond) and thus assert row and column requests. This prevents the peripheral circuits from communicating a different pixel address. It would be beneficial to prevent a pixel from accessing the bus during pixel reset.

7.3 Requirements

Additional to the most obvious requirement 'as fast as possible', our dynamic pixel circuits impose a few additional requirements and constraints:

- Preventing a pixel from requesting during the time the pixel differentiator is reset is crucial to increase throughput.
- Keep transistor count in the pixel as low as possible.
- Pixels with very low thresholds should not be able to completely block the communication by constantly accessing the bus. In the DVS128, the pixel with the lowest threshold limits contrast sensitivity.
- Our pixels do not have positive feedback after crossing threshold. Thus they can go back below threshold after asserting a row request. We need to ensure that the periphery does not get stuck if there is a row request but no column request.

Keeping these requirements in mind and inspired by Boahen's word serial AER circuits [31], we developed our own word serial AER sender circuits to be able to scale up our arrays without sacrificing per-pixel bandwidth.

7.4 Word serial AER summary

The AER interface presented here is adapted from Boahen's word serial circuits [31]. Unlike the AER circuits used in the DVS128, it needs more than one handshake cycle to transmit one event. Although it may sound counterintuitive, this can speed

up the communication. Additionally, it cuts the number of address pads needed almost in half.

Very simplified, the protocol works as follows. Row arbitration is similar to the DVS128, but as soon as one row is selected, the row address is transmitted, at the same time the state of all pixels (requesting or not requesting) in this row is latched and then the column addresses of the active pixels are transmitted sequentially. This will be slower than original AER if there is only a single event in a row, but will be more efficient at high loads, because signal transitions in the column direction can happen in parallel and column arbitration is simplified. In the original AER protocol, handshaking with the receiver can only start when row and column arbitration is finished and both row and column addresses are driven. In the word serial case, communication of the row address can start earlier, negating the disadvantage of the necessary additional handshake cycle. Boahen's article [31] describes in more detail why the word serial protocol provides more efficient communication than the previous AER circuits.

So then the question remains why we do not directly use Boahen's scheme. The following list states our reasons:

- Our pixels have two polarities, some modification of the scheme is needed in any case.
- In Boahen's circuits, the pixels hold on to the row request until they are reset. In our case, resetting the differentiator of the dynamic pixels can take a considerable amount of time, during which the pixels in the DVS128 hold on to the row request, blocking the communication. If we used Boahen's scheme directly, the same would be true. Thus, using Boahen's circuits without modification violates our first and most crucial requirement to improve communication bandwidth.
- Boahen's pixel circuit can do without a state-holding element, if the event signal inside the pixel is a fast and clean downward transition [31]. In our case, this can not be guaranteed, because the pixel might cross the threshold arbitrarily slow and there is no positive feedback. If we use directly Boahen's scheme, we would need state-holding elements in the pixel, increasing area demand considerably. We rather increase the complexity of the periphery so as to keep the pixel complexity as low as possible.

The periphery needs added complexity compared to Boahen's word serial circuits, because our pixels do not have positive feedback after crossing threshold, so they do not latch events. It is possible that a pixel goes back below threshold after it successfully asserted a row request. We need to ensure that the periphery does not get stuck if there is a row request but no column request. How we deal in this is described in section 7.7. We avoid having a latch in the pixel to decrease transistor count and power consumption.

The main parts of our AER circuits are a fair arbiter [31] for row arbitration, an asymmetrical column arbiter, a column of "Endpixels" and a main state machine

which handles the communication with all the blocks and the receiver. Fig. 7.2 shows a block diagram.

We used LogSpice developed by John Arthur and Paul Merolla at Kwabena Boahen's lab in Standford [11] to verify and simulate this new AER scheme. See appendix C for a quick tutorial on LogSpice. For a general introduction to asynchronous digital circuits and the notation, refer to the article by Martin and Nyström [129].

For off-chip communication, our implementation of the AER protocol uses single rail bundled data as opposed to more advanced and safer transmission protocols like dual-rail or one-hot encoding [129], because we want our chips to be able to interface to micro-controllers. The additional complexity of advanced encodings will likely overstrain low-power micro-controllers. All the signals on the chip are single-rail.

7.5 Top Level Signals

The following list shows the signals that are relevant to the top level (Fig. 7.2). Some of the signals are active low, but for simplicity all signals except *nReset* are used in active high form throughout this chapter.

Creq, Cack Chip request and acknowledge.

Rrow, Arow Request and acknowledge from the row arbiter to the main state machine, *Rrow* is a wired-OR with keeper (see section 7.9).

Rcol, Acol Request and acknowledge from the column arbiter to the main state machine, *Rcol* is a wired-OR with keeper.

Lcol When this signal is active, the column state machines latch active *Rxcol* signals and assert corresponding *Rxarb*. *Rxcol* signals arriving after *Lcol* is taken away are not serviced.

XSel Indicates whether row or column addresses are communicated. This signal is used for the address multiplexer.

Rxcol, Axcol Column Request and Acknowledge signals between pixel array and column arbiter. *Rxcol* is implemented as a wired-OR with a keeper.

RxcolG Wired OR of all column request lines, used to make sure that all these lines are properly reset after a communication cycle. Its pull-down bias is switched on only when necessary to avoid unnecessary power consumption.

Rxarb, Axarb Request and Acknowledge signals between column arbiter state machines and asymmetric arbiter tree.

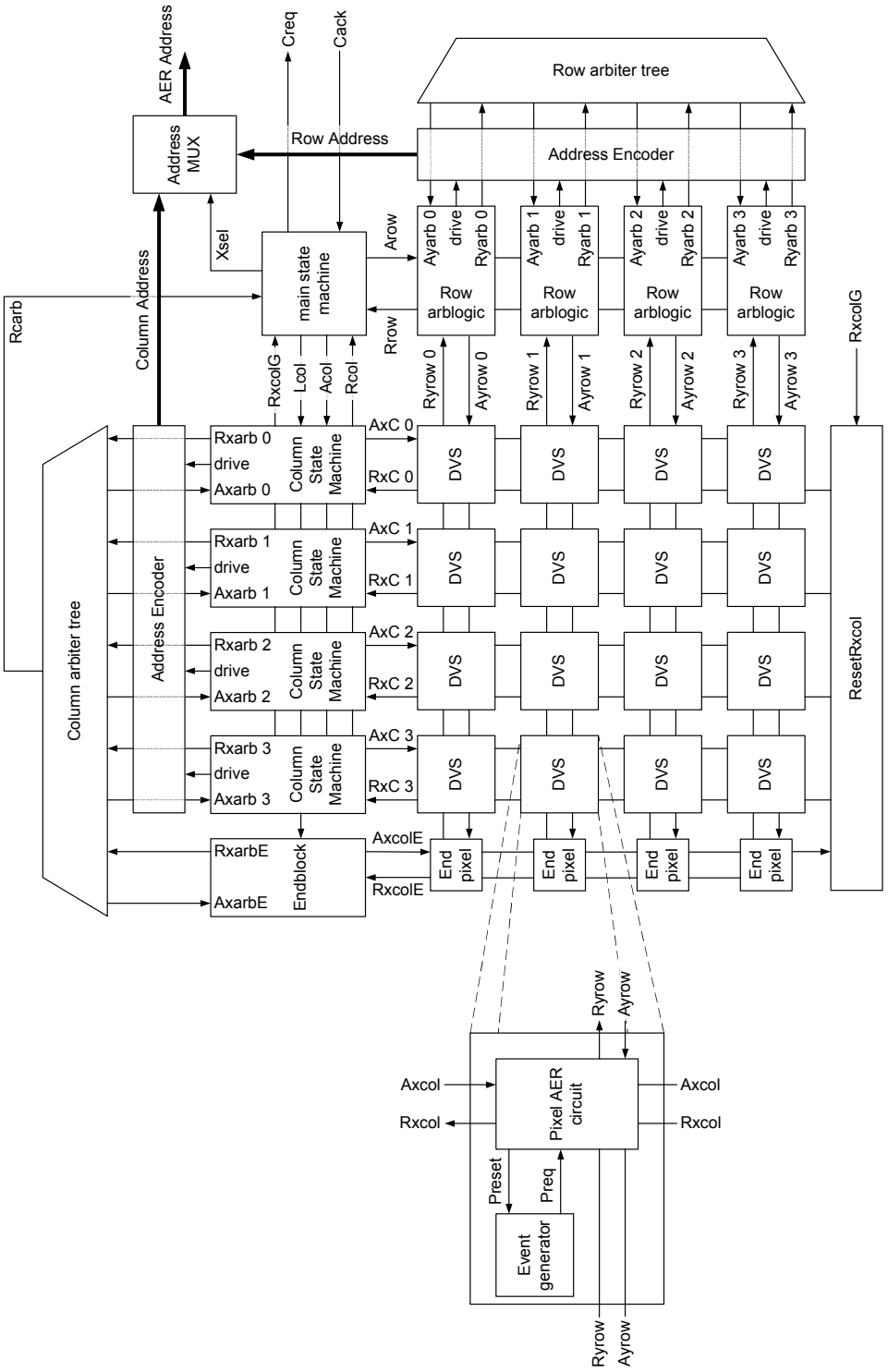


Figure 7.2: Word serial AER block diagram for a 4×4 pixel array. The signal names *Axcol* and *Rxcol* have been shortened to *AxC* and *RxC*.

Rcarb is the output of the asymmetric column arbiter tree and serves as indicator whether at least one column is active. It serves as a logic OR between all Rxarb.

Ryrow, Ayrow Row Request and Acknowledge signals between pixel array and row arbiter. *Ryrow* is implemented as a wired-OR with a keeper.

Ryarb, Ayarb Request and Acknowledge signals between row arblogic and row arbiter tree.

nReset Used to reset all the state machines as well as all the pixels by means of forcing all the *Axcol* and *Ayrow* simultaneously.

In the real implementation, *Rxcol* is split into *RxcolOn* and *RxcolOff* to account for event polarity, but for simplicity we describe the scheme with only one signal here.

7.6 Sequence for transmitting events

When a pixel crosses threshold, it requests first in the row direction to the row arbiter. The acknowledge from the row arbiter enables the column request in the pixel (either *RxcolOn* or *RxcolOff*, depending on the polarity of the event). Meanwhile the row arbiter drives the row address and asserts *Rrow* to the main state machine, which starts handshaking with the off-chip receiver to transmit the row address.

As soon as the receiver activates *Cack*, the main state machine asserts *Lcol*. Now the column state machines with corresponding active *Rxcol* signals latch these requests and the polarity. These state machines assert their *Rxarb* signals. The output of the column arbiter tree *Rcarb* becomes active when at least one of the *Rxarb* signals is active.

After finishing the row address transmission, the main state machine asserts *Arow* and *XSel*. It will take away *Lcol* when *Cack* is taken away and *Rcarb* is active. This means at least one pixel has asserted its column request (exception see section 7.7).

In the mean-time the leftmost active column state machine will get the acknowledge from the asymmetric arbiter tree and will therefore drive its column address and pull *Rcol* high. The main state machine then handshakes with the receiver to transmit the corresponding column address and after completing asserts *Acol*.

The column state machine will take away *Rxarb* and reset *Rcol*, and the next column state machine will get acknowledge from the arbiter tree and the corresponding address will be transmitted.

Subsequently all column addresses will be transmitted until all column state machines are back in their idle state. Because no *Rxarb* will be active, *Rcarb* will become inactive as well, signaling the main state machine that all column addresses

are serviced. The main state machine goes back to its idle state as soon as *RxcolG* is inactive, which means that all *Rxcol* signals have been reset.

Figure 7.3 shows example time lines from LogSpice for the most important signals. In both Figures 7.3(a) and 7.3(b), the pixels cross threshold at exactly the same time points (220, 1000, 1100, 1760 and 1990, time-scale is arbitrary). The difference in the two plots arise from different delays in the signals, which results that in Fig. 7.3(a), the events happening at time points 1760 and 1990 are transmitted in a single burst, while in 7.3(b), the event at 1990 arrives to late to be transmitted in the same burst as the event at 1760. The first event at time point 220 is a row-only event, where a pixel asserts row request, but no pixel asserts column request.

7.7 The need for the Endpixels and the Endblock

If the pixel array strictly followed the four-phase handshake protocol, Endpixels and Endblock would not be needed. But because the pixels do not latch events, they can go back below threshold after successfully activating the row request and therefore not assert a column request. This means that it is possible that no pixel in a row asserts a column request when the row acknowledge is returned.

That the handshaking circuits do not get stuck in such a situation, we include an Endpixel at the end of each row. These are basically pixels which always make a column request when the row acknowledge is asserted. Their column request goes to the Endblock, which is a simplified column state machine. It also requests to the arbiter tree but its address is not transmitted. Because it also requests to the arbiter tree, *Rcarb* becomes active even if no pixel in the array asserts a column request. This makes the main state machine take away *Lcol* and finally go back to its idle state when *RxcolE* is reset and thus *RxcolG* becomes inactive.

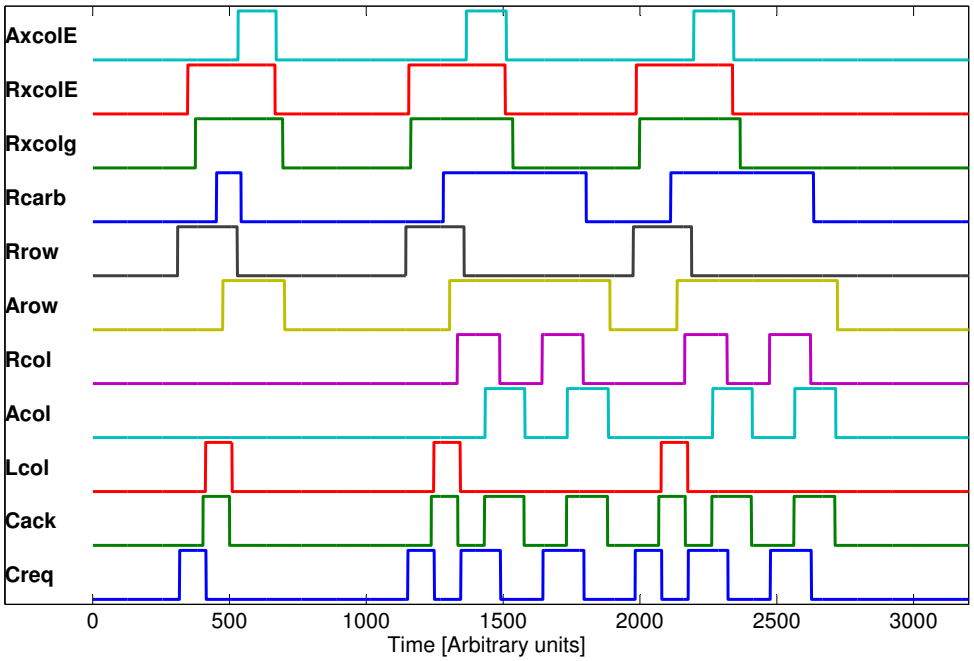
7.8 State-holding elements

The main state machine, the Endblock, the Endpixels as well as the column state machines (which serve as interface between the array and the column arbiter) are implemented as asynchronous state machines with four states. States are encoded using Gray codes. The state-holding elements in the state machines are double cross-coupled NOR Set-Reset gates illustrated in Fig. 7.4. The corresponding truth table is shown in table 7.1.

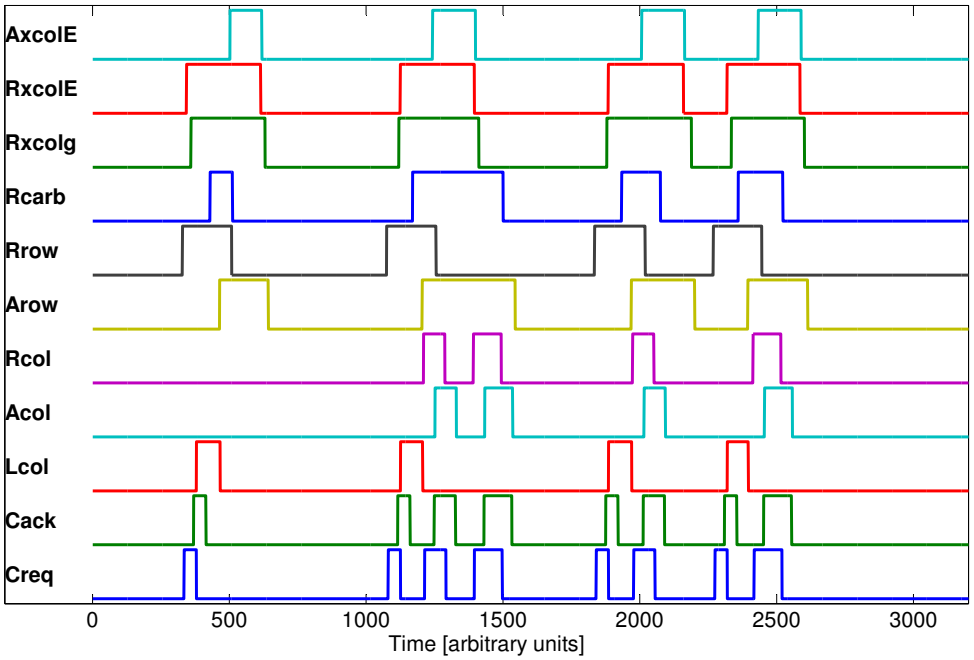
Fig. 7.5 illustrates schematically the state machines with two state-holding elements. In all the following descriptions, the two state variables are called FF1 and FF2, the idle state is $FF2 = 0$ and $FF1 = 0$, transitions are

$$00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00,$$

where 01 stands for $FF2 = 0$ and $FF1 = 1$.



(a)



(b)

Figure 7.3: AER signal time lines

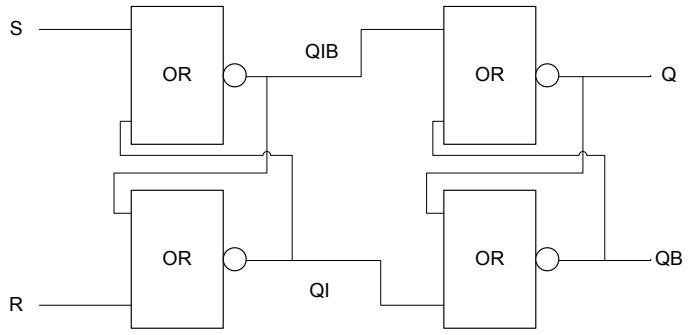


Figure 7.4: NOR Set-Reset Gate

S	R	QI	QIB	Q	QB
0	0	QI	QIB	Q	QB
0	1	0	1	0	1
1	0	1	0	1	0
1	1	0	0	Q	QB

Table 7.1: Truth table for double NOR SR latch

The row arblogic, the ResetRxcol-block as well as the arbiter trees use staticizers as state-holding elements. Staticizers use a strong forward inverter and a weak feedback inverter to store the current state. The feedback inverter has to be strong enough to hold the state and weak enough so it can be overcome by the inputs. Fig. 7.6 illustrates the concept.

A difference between using Set-Reset gates and staticizers as state-holding elements is that Set-Reset gates need full logic gates to produce the set and reset signals, while a staticizer only needs a pull-down path for the set signal and a pull-up path for the reset signal, which results in more compact logic. The disadvantage of staticizers is that the transistors of these paths and the feedback inverter have to be carefully sized to make sure the state is held and that the input can still overcome the feedback inverter. For the Set-Reset gate, transistor sizing affects switching speed, but it is functionally less critical.

7.9 Keepers

In the DVS128, all the wired OR request signals (from the array to the arbiters and also the chip request which comes from the column arbiter) have static pull-ups (or pull-downs, depending on the polarity) that reset the request line after event communication. To achieve sufficiently fast hand-shake cycle times, these static pull-ups have to be turned on relatively hard, which increases power consumption

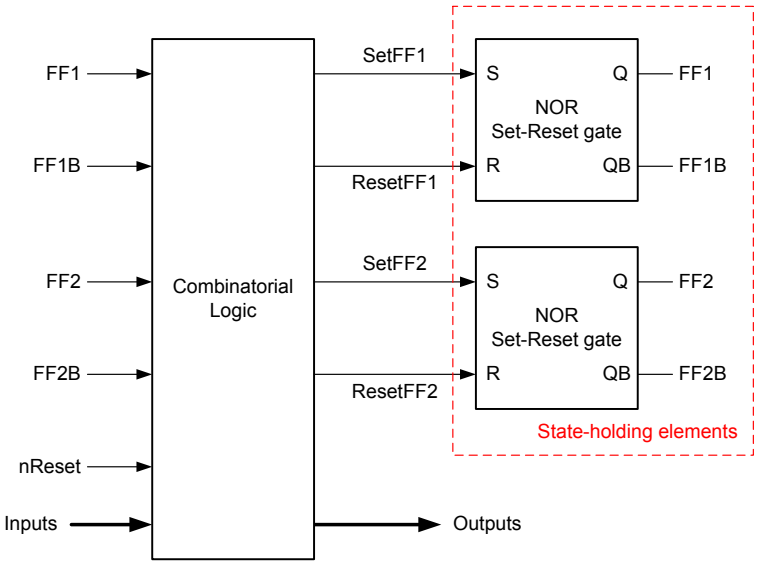


Figure 7.5: Schematic illustration of a state machine

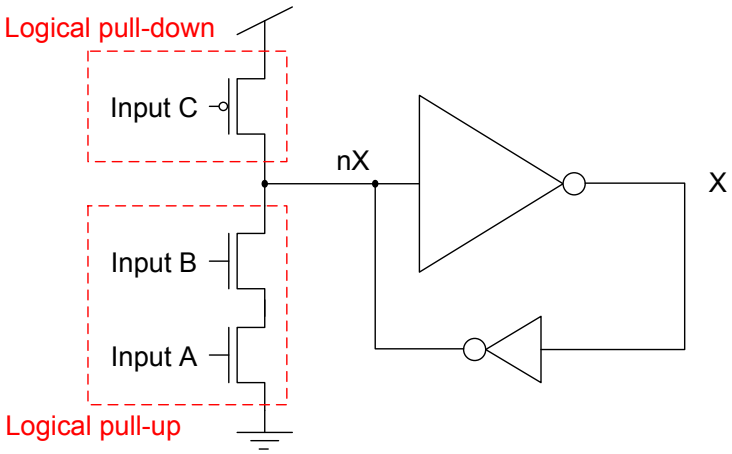
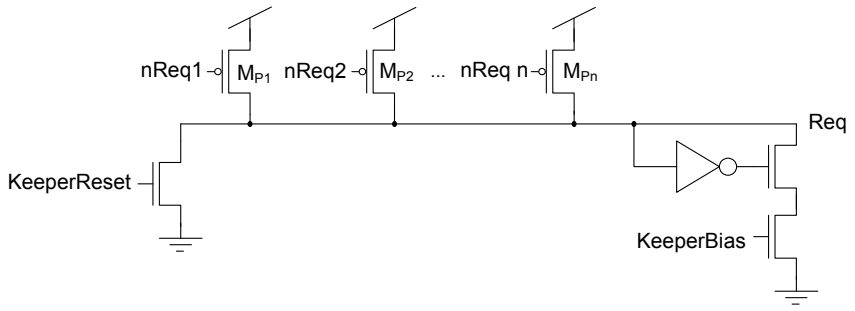
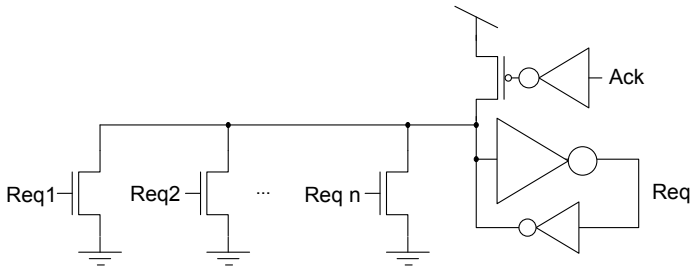


Figure 7.6: Staticizer. The logical pull-up path sets the staticizer (this is not a typo in the illustration, the logical pull-up actually pulls down the inverted output signal), the logical pull-down clears the output signal. This example staticizer is set when *Input A* and *Input B* are true simultaneously, it is reset when *Input C* is false. Of course one has to make sure that the pull-down path and the pull-up path are not active at the same time.



(a) Our keeper circuit. The inputs to the Wired-OR are active low, the output is active high.



(b) Keeper circuit used by Boahen. This circuit has active high inputs and an active high output.

Figure 7.7: Wired-OR request line with keeper.

when active pixels are pulling against the static pull-up.

To reduce power consumption, we use keepers (Fig. 7.7(a)) instead of static pull-ups. A keeper keeps the request line in its inactive state as long as no pixel is pulling on the request line. If a pixel pulls sufficiently, the keeper is switched off and the pixel pulls up the request line without opposition. Once the keeper is switched off, the combined leakage current of the pull-up transistors M_{P1} to M_{Pn} is enough to keep the request line in its active state, even if no pixel is pulling anymore.

The keepers increase the complexity of the circuits, because they have to be actively reset. From this follows that handshaking with the pixels is not a full four-phase hand-shake anymore, because the pixels do not take the request away. Reset of the keepers has to be synchronized with the reset of the pixel, a pixel still pulling down the request line while the request line is already being reset results in excess power consumption.

Compared to the keepers used by Boahen, which are based on staticizers as illustrated in Fig. 7.7(b), our keepers have the advantage that their keeping-strength can be adjusted. The keeping-strength of a staticizer is fixed by design. Additionally, Boahen resets the keeper with the acknowledge signal and does not ensure

that no pixel is asserting request anymore. This leads to excess power consumption. Boahen's circuit relies on the reset pFET being weaker than the nFETs to ensure that the request is high until all active pixels are reset.

7.10 Detailed description of blocks

The way the asynchronous state machines are implemented introduces some implicit timing assumptions. The delay from an input signal to the two state variables must be similar as well as the delay from the two state variables to an output signal should be similar to avoid glitches. Similar here means that the wire delay difference must not be bigger than a gate delay. This assumption however is very reasonable, as the state variables are local and therefore located very close to each other, i.e. within a few micrometers.

Most of the implemented blocks include a reset signal *nReset*, for simplicity it is not included in the Hand-Shaking Expansion notation.

The syntax of the production rules given in this chapter do not follow the syntax from Martin [129], but the syntax from LogSpice (see Appendix C). For each signal, a logical pull-up (LPU) and a logical pull-down (LPD) path with the following syntax is given.

```
LPUname signal condition LPU
LPDname signal condition LPD
```

If the `condition` of the LPU is true, `signal` becomes active, if the `condition` of the LPD is true, `signal` is reset to inactive. `condition` can be any logical combination of input signals.

Unfortunately LogSpice does not allow direct implementation of a WiredOR with a keeper, which would be several logical pull-ups and a single pull-down. Therefore a slightly more complicated description is necessary with full set and reset signals. In practice the reset of a keeper can usually be implemented as just a pull-down path like transistor M_r in Fig. 7.7.

7.10.1 Pixel AER circuits

If a pixel crosses threshold, it activates first its row request *Ryrow*. As soon as the row acknowledge *Ayrow* is active, the pixel stops pulling on *Ryrow* (it can not reset *Ryrow*, this is done in the periphery by the Endpixel) and activates the column request (either ON or OFF depending on the polarity of the event). If the pixel gets the column acknowledge, it releases the column request and resets itself. If the pixel does not get the column acknowledge (it is too late to be serviced in this communication cycle), the pixel activates the row request again as soon as *Ayrow* is inactive. Fig. 7.8 shows the pixel AER block with inputs and outputs.

The Hand-Shaking Expansion (HSE) of the pixel AER circuits, following the notation of Boahen [31], is as follows

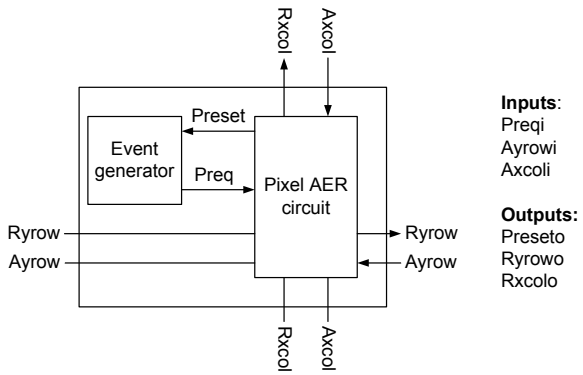


Figure 7.8: Pixel AER inputs and outputs

```
*[[Preqi & ~Ayrowi]; Ryrowo+; [Ayrowi]; Rxcolo+; [Axcoli];
  Preseto+;  [~Preqi & ( ~Ayrowi | ~Axcoli ) ]; Preset-;]
```

Note that *Ryrow* and *Rxcol* are set by the pixels, but the pixels can not reset these wired-OR signals.

This leads to the following production rules:

```
LPUryrow Ryrowo {~Ayrowi & ~Preseto & Preqi} LPU

LPUrxcol Rxcolo {~Preseto & Preqi & Ayrowi} LPU

LPUpreset Preseto { Ayrowi & Axcoli} LPU
LPDpreset Preseto { ~Ayrowi | ~Axcoli} LPD
```

Ryrow and *Rxcol* only have a pull-up path in the pixel. As soon as the condition of the pull-up is not fulfilled anymore, the pixel stops pulling up these signals. Pull-down of *Ryrow* and *Rxcol* is handled at the periphery.

The pull-down path of *Preset* is implemented as a bias current discharging a capacitor, as illustrated in Fig. 7.9(a). The bias has to be adjusted to be weak enough so that *Preset* is active long enough to fully reset the pixel. If this dependence on a bias current is unacceptable, an additional transistor can be included to implement the following logical pull-up rule:

```
LPDpreset Preseto {~Preqi & (~Ayrowi | ~Axcoli)} LPD
```

This additional transistor can be used to enable the bias current only when the event is cleared.

The actual transistor level implementation (ignoring event polarity) along with the transistor level circuits of the DVS128 and Boahen's AER circuits is shown in Fig. 7.9. Transistor count is 12 transistors for our new AER scheme, six for the DVS128 and 12 for Boahen's scheme. However, if we add the transistors for the staticizers for signals W and P, the total transistor count of Boahen's scheme is 18.

We slightly modified the reset and refractory circuit of our new scheme compared to the DVS128, because at low refractory bias currents, switching of *Axcol* can couple to *nPreset*, which in turn affects the differentiator. This modification is independent of the word serial AER scheme and adds two transistors. Thus the word serial scheme adds just four transistors compared to the DVS128.

The pixel improves the AER circuits from the DVS128 to speed up the communication, because the pixel does not claim the bus until the differentiator is reset. As soon as both row and column acknowledge are asserted, both row and column request are released by the pixel reset signal. Additionally, the row request is released as soon as row acknowledge is asserted.

7.10.2 Main state machine

The main state machine orchestrates the communication with the row and column arbiters and handles the communication with the off-chip receiver. Communication is started with a request from the row arbiter. The main state machine then handshakes with the off-chip receiver to transmit the row address. After completing this handshake, the main state machine asserts *Arow* and *XSel*. Subsequently the column addresses of active pixels are transmitted. *Rcarbi* acts as OR between all unserviced column requests and stays high as long as there are column addresses to transmit. *Rcarbi* going low tells the state machine that all column addresses have been transmitted. Fig. 7.10 shows the inputs and outputs of the main state machine, Fig. 7.11 illustrates the state transitions in a graphic way.

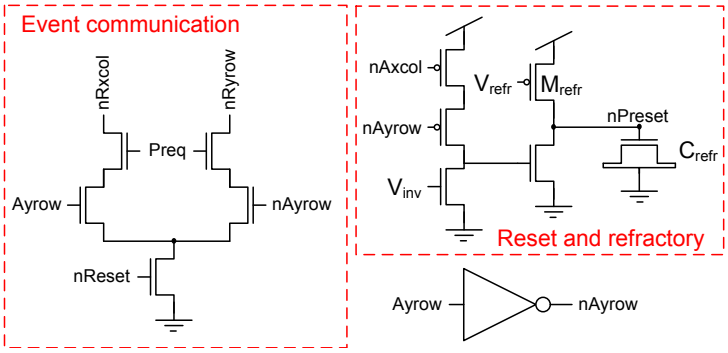
The signal *RxcolGBiaso* switches on the pull-down bias for the wiredOR for *RxcolG*.

Hand-Shaking Expansion

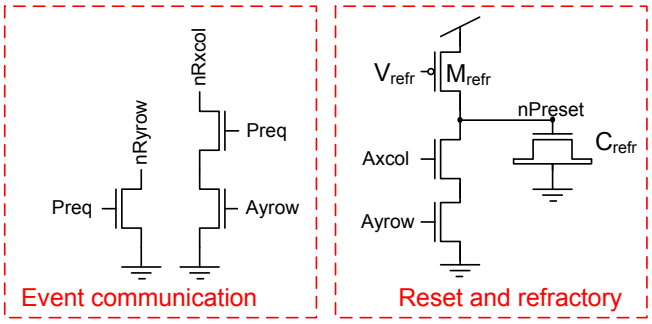
```
*[[Rrowi]; Creq+; [Cacki]; FF1+; Creq-, Lcolo+, RxcolGBiaso-;
  [Rcarbi]; FF2+, XSel+; Arowo+; [~Cacki]; FF1-; Lcolo-, RxcolGBiaso+;
  [ [Rcoli]; Creq+; [Cacki]; Acolo+; [~Rcoli]; Creq-; [~Cacki]; Acolo-;
    ||
    [~Rrowi & ~RxcolGi & ~Rcarbi & ~Rcoli & ~Cacki]; FF2-, XSel-;
    Arowo-; ]]
```

Production rules

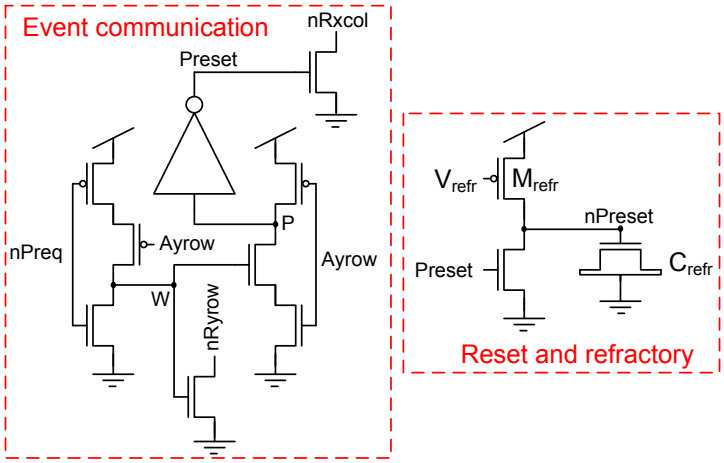
```
LPUFF1 FF1 {(~FF2 & Cacki ) & nReset} LPU
LPDFF1 FF1 { ( FF2 & ~Cacki ) | ~nReset} LPD

LPUFF2 FF2 {( FF1 & Rcarbi ) & nReset} LPU
LPDFF2 FF2 { ( ~FF1 & ~Rrowi & ~RxcolGi
           & ~Rcarbi & ~Rcoli & ~Cacki ) | ~nReset} LPD

xarow Arowo FF2 ~nReset OR2
```



(a) Transistor level circuit of our AER scheme.



(b) Transistor level circuit of the DVS128.



(c) Boahen's transistor level circuit.

Figure 7.9: Pixel AER transistor circuit comparison, ignoring event polarity.

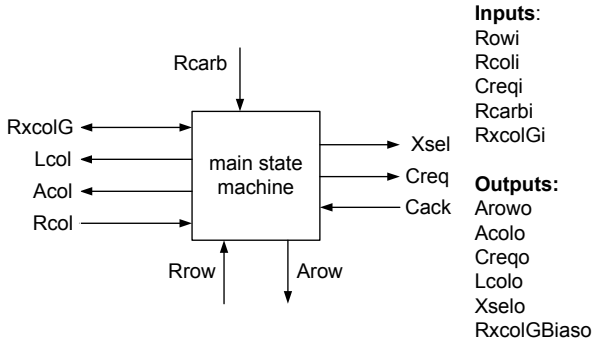


Figure 7.10: Inputs and outputs of the main state machine

```
LDUlcol Lcolo { FF1 } LPU
LPDlcol Lcolo {~FF1 } LPD

LDUxsel XSelo { FF2 } LPU
LPDxsel XSelo {~FF2 } LPD

LPUAcol Acolo {~FF1 & FF2 & Cacki} LPU
LPDAcol Acolo { FF1 | ~FF2 | ~Cacki} LPD

xCreq Creqo {(Rrowi & ~FF1 & ~FF2)} {(Rcoli & ~FF1 & FF2)} OR2

LPURxcolgbias RxcolGBiaso {~FF1 | ~nReset} LPU
LPDRxcolgbias RxcolGBiaso {FF1 & nReset} LPD
```

7.10.3 Row arblogic and arbiter

The row arblogic blocks serve as interface between the pixel array, the arbiter tree, and the main state machine. If a pixel in the row activates the row request line *Ryrow*, the row arblogic asserts *Ryarb*, the request to the arbiter tree. When the arblogic gets the acknowledge from the arbiter tree (*Ayarb*) and the acknowledge from the main state machine *Arow* is not active, it asserts the acknowledge to the row of pixels *Ayrow*, the request to the main state machine *Rrow* and drives its row address. When the main state machine asserts *Arow*, the arblogic takes *Rrow* and *Ryarb* away. *Ayrow* is taken away when also the row request *Ryrow* is gone.

The row arblogic is almost equal to the circuits presented by Boahen [76], the only difference being that we include a *nReset* input, which asserts *Ayrow*.

The row arbiter tree is equal to the one used in the DVS128 [66], which is a fair arbiter [31]. The fair arbiter remembers which of its two inputs was active last, and in case both inputs are requesting, the input that has not been serviced for a longer time will be serviced first. A row that was just serviced will only be serviced again

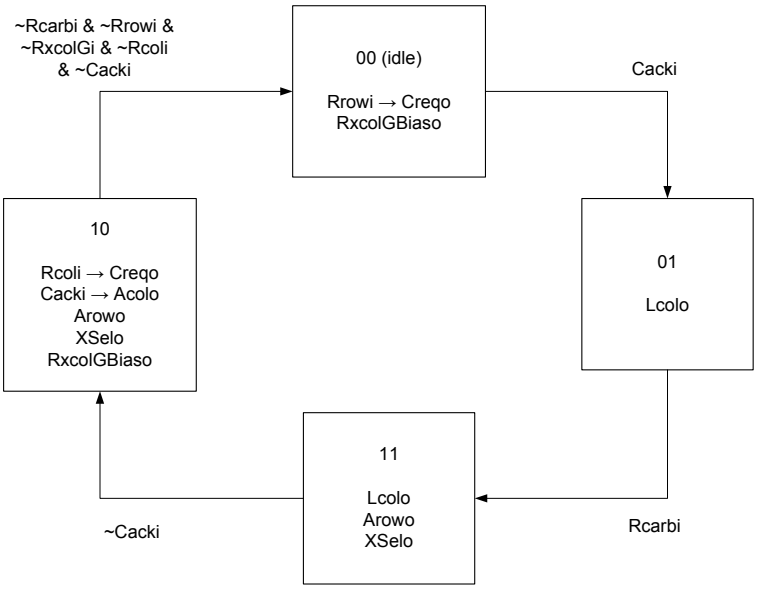


Figure 7.11: State diagram of main state machine. Output signals which are not mentioned in a state are inactive. Arrows signify state transitions, the signal name written next to the arrow is the condition for the transition. A tilde in front of a signal means negation, & stands for a logical AND. *Rrowi* → *Creqo* means that in the corresponding state, the output *Creqo* is active if the input *Rrowi* is active.

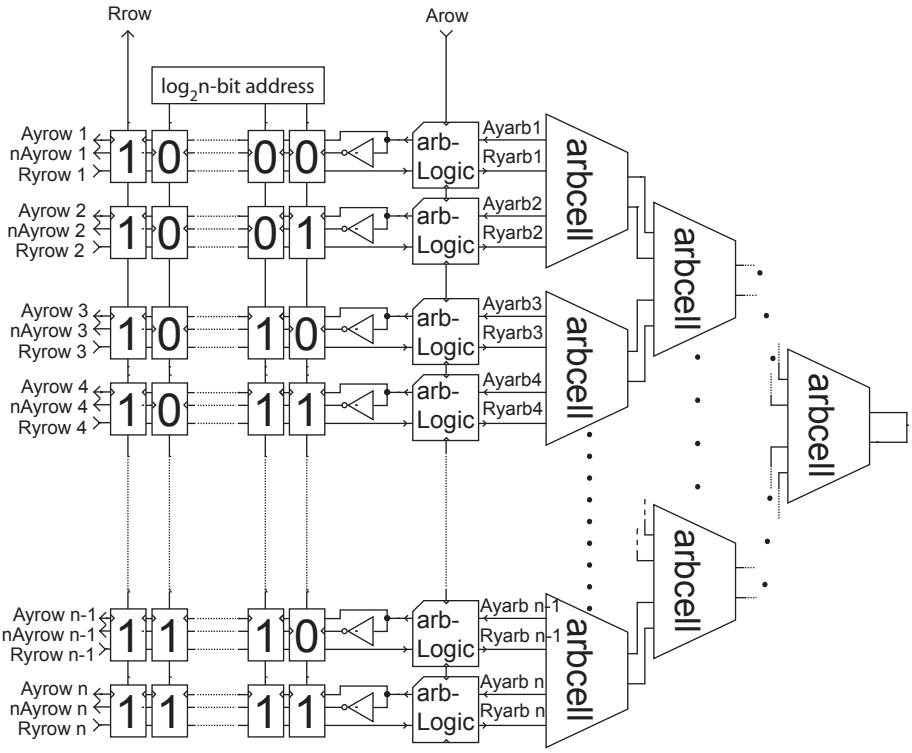


Figure 7.12: Row arbiter schematic, adapted from [39].

once all other pending requests have been serviced. The full arbiter with arblogic, address encoder and arbiter tree is illustrated in Fig. 7.12.

Hand-Shaking Expansion

```
*[[Ryrowi]; Ryarbo+; [Ayarbi & ~Arowi]; Ayrowo+; Rrowo+;
  [Arowi]; Rrowo-, Ryarbo-; [~Ayarbi & ~Ryrowi]; Ayrowo-;]
```

Production rules

```
LPUryarb Ryarbo { Ryrowi & ~Ayrowo} LPU
LPDryarb Ryarbo { Ayrowo & Arowi } LPD

LPUayrow Ayrowo {(~Arowi & Ryrowi & Ayarbi) | ~nReset } LPU
LPDayrow Ayrowo { ~Ryrowi & ~Ayarbi & nReset } LPD

LPUrrow Rrowo {~Arowi  & Ayrowo & Ryarbo} LPU
LPDrrow Rrowo Arowi LPD
```

7.10.4 Column state machine

The column state machine serves as an interface between the pixel array, the column arbitration tree, and the main state machine. Its sequence of action is very similar to the row arblogic, with the difference that it only services column request that are active during the time the input signal *Lcol* is active.

The state machine leaves its idle state only when *Rxcol* is activated while *Lcol* is high. If *Rxcol* arrives when *Lcol* is inactive again, the request is not serviced. *Rxcol* is the OR between *RxcolOn* and *RxcolOff*, the polarity is latched by a D-type flipflop clocked by *FF1*. Fig. 7.13 shows the input and output signals of the column state machine, Fig. 7.14 illustrates the state transitions in a graphic way.

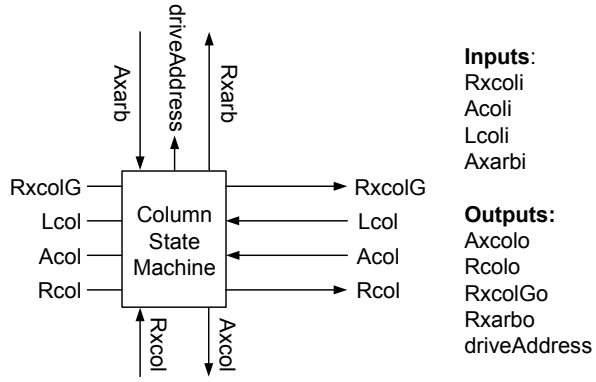


Figure 7.13: Inputs and outputs of the column state machine.

Hand-Shaking Expansion

```
*[[Rxcoli]; RxcolG+; ||
  [[Rxcoli & Lcol]; FF1+; Axcolo+, Rxarbo+;
  [ [~Rxcoli]; Axcolo-; ||
    [Axarbi & ~Acoli]; FF2+; Rcolo+; [Acoli];
    FF1-, Rcolo-; Rxarbo-; [~Acoli & ~Rxcoli]; FF2-;]] ]
```

Production rules

```
LPUFF1 FF1 { (Rxcoli & Lcoli & ~FF2) & nReset} LPU
LPDFF1 FF1 { (Acol & FF2) | ~nReset} LPD

LPUFF2 FF2 { Axarbi & ~Acoli & FF1 & nReset} LPU
LPDFF2 FF2 { (~Acol & ~Rxcoli & ~FF1) | ~nReset } LPD

LPURcol Rcolo { FF1 & FF2 & ~Acoli} LPU
LPDRcol Rcolo Acoli LPD
```

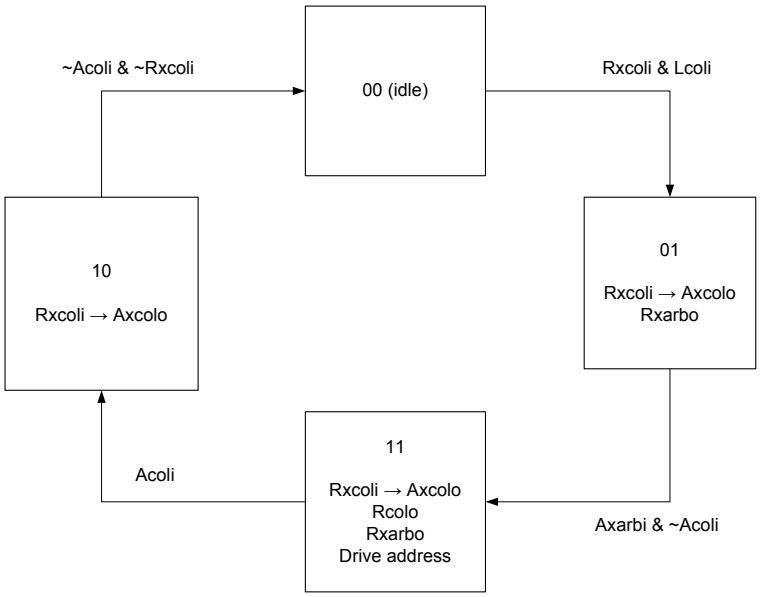


Figure 7.14: State diagram of column state machine.

```
xAxc Axc { FF1 | FF2 } Rxcoli AND2
xAxcol Axcolo Axc ~nReset OR2

LPUrxarb Rxarbo FF1 LPU
LPDrxarb Rxarbo ~FF1 LPD

LPUrxcolg RxcolG Rxcol LPU
```

The signal *RxcolG* is a wired OR between all *Rxcol*. Every column state machine can set *RxcolG*, but only the main state machine can enable the bias to reset *RxcolG*.

7.10.5 Asymmetric Column Arbiter

The column arbiter is built as a binary tree of two-input arbiter cells (Fig. 7.15) like the fair arbiter [76]. However, we use a simplified arbiter cell for column arbitration. As all active pixels in one row are transmitted in the same burst, the order of the events does not matter, fairness is therefore not necessary. Simplifying the arbiter cell can speed up arbitration, because there are less gate-delays per arbiter cell.

Port one (back1 and breq1) has higher priority than port two (back2 and breq2). Port one will be selected if both inputs are requesting at the moment the acknowledge comes from the top input. If however the lower priority input already is acknowledged when the higher priority input request becomes active, the lower priority port will not lose the acknowledge and port one has to wait until port two is serviced.

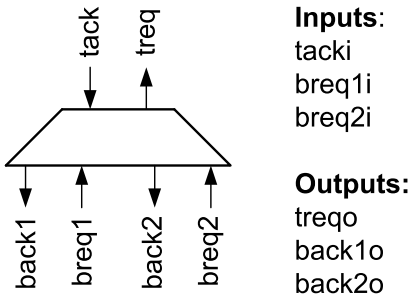


Figure 7.15: Column arbiter cell

Production rules

```
LPUt treqo { breq1i | breq2i} LPU
LPDt treqo {~breq1i & ~breq2i} LPD

LPUba1 back1o { breq1i & tacki & ~back2o} LPU
LPDba1 back1o {~tacki | ~breq1i } LPD

LPUba2 back2o { breq2i & tacki & ~back1o & ~breq1i } LPU
LPDba2 back2o { ~tacki | ~breq2i | back1o } LPD
```

7.10.6 Endblock

The Endblock is a simplified column state machine that handshakes with the Endpixels. The difference to the column state machine is that it does not have an address encoder and it does not handshake with the main state machine. It goes back to idle state when it gets acknowledged from the arbiter tree. Fig. 7.16 shows the inputs and outputs, Fig. 7.17 illustrates the state transitions in a graphic way.

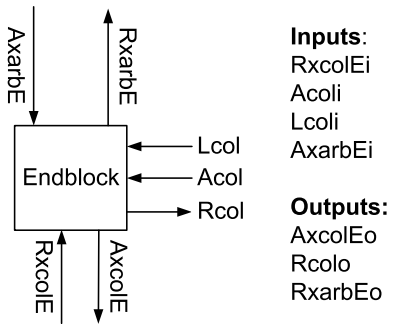


Figure 7.16: Endblock inputs and outputs

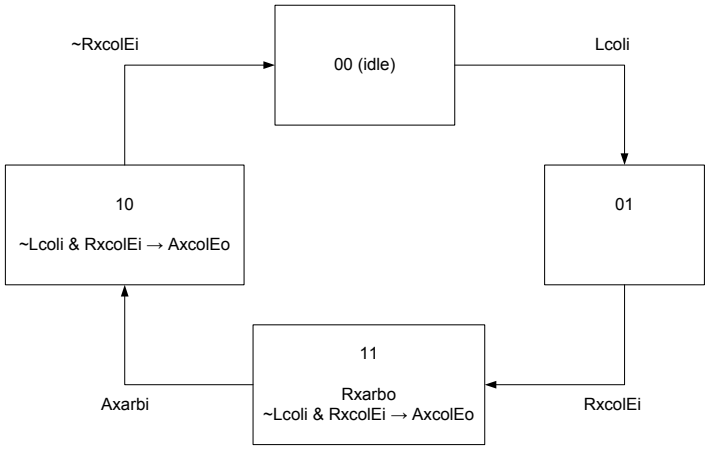


Figure 7.17: Endblock state diagram

Hand-Shaking Expansion

```
*[[Lcoli]; FF1+; [RxcolEi]; FF2+; RxarbEo+;
  [ [~Lcoli];  AxcolEo+; ||  [AxarbEi]; FF1-; RxarbEo-; ]
  [~RxcolEi]; FF2-, AxcolEo-; ]
```

Production rules

```
xrxarb RxarbEo FF1 FF2 AND2

xaxcole AxcolEo FF2 RxcolEi ~Lcoli AND3

LPUFF1 FF1 { (Lcoli & ~FF2) | ~nReset } LPU
LPDFF1 FF1 { FF2 & AxarbEi & nReset } LPD

LPUFF2 FF2 { FF1 & RxcolEi & nReset } LPU
LPDFF2 FF2 { (~FF1 & ~RxcolEi) | ~nReset } LPD

LPDrcol Rcolo Acoli LPD
```

In the Endblock, *Acol* resets the keeper of *Rcol*.

7.10.7 Endpixels

The Endpixels are a special column of pixels to the left of the pixel array. These special pixels do not have analog circuits, they are only used for the communication cycle as described in section 7.7. Whenever the row acknowledge signal becomes active, the Endpixel activates its column request *RxcolE*. When also the Endpixel

column acknowledge *AxcolE* is active, it resets the row request signal *Ryrow* to its inactive state by means of *ResetRyrow*.

Fig. 7.18 shows the inputs and outputs, Fig. 7.19 illustrates the state transitions in a graphic way.

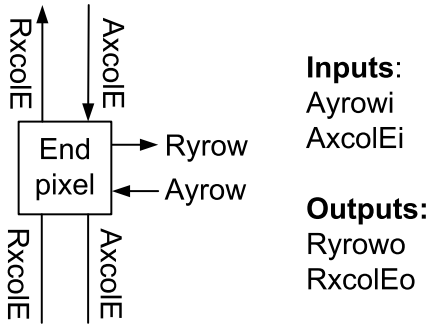


Figure 7.18: Inputs and outputs of the Endpixel

Hand-Shaking Expansion

```
*[[Ayrowi]; FF1+; RxcolEo+; [AxcolEi] FF2+; ResetRyrowo;
  [~Ayrowi]; FF1-; RxcolEo-, ResetRyrowo-; [~AxcolEi]; FF2-;]
```

Production rules

```
LPUResRyrow ResetRyrowo {~nReset | ( FF1 & FF2 ) } LPU
LPDResRyrow ResetRyrowo { nReset & ( ~FF1 | ~FF2) } LPD

LPUrxcolE RxcolEo { FF1 & ~FF2 } LPU
LPDrxcolE RxcolEo {~FF1 & FF2 } LPD

LPUFF1 FF1 { Ayrowi & ~FF2} LPU
LPDFF1 FF1 {~Ayrowi & FF2} LPD

LPUFF2 FF2 { AxcolEi & FF1} LPU
LPDFF2 FF2 {~AxcolEi & ~FF1} LPD
```

7.10.8 ResetRxcol

This block is responsible for resetting all column requests. The signal *ResetRxcol* is brought along the bottom of the array. *Rxcol* can not be reset with *Axcol*, because *Axcol* might not be active even though *Rxcol* is, because the pixel asserted *Rxcol* too late and the column state machine did not service the event in this burst.

ResetRxcol is asserted until the wired OR between all the requests *RxcolG* becomes inactive, which means all the requests have been cleared.

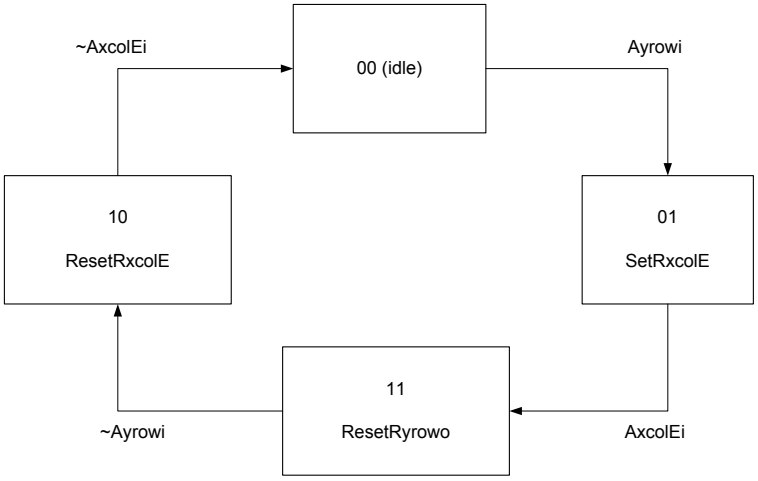


Figure 7.19: Endpixel state diagram

The ResetRxcol-block with inputs and outputs is illustrated in Fig. 7.20.

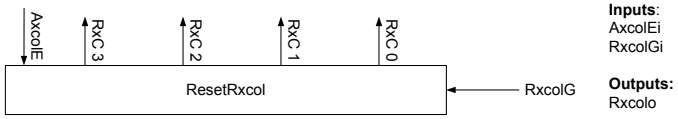


Figure 7.20: ResetRxcol inputs and outputs

Hand-Shaking Expansion

```
*[[AxcolEi]; x+; [~AxcolEi]; ResetRxcolo+;
  [~RxcolGi]; x-, ResetRxcolo-; ]
```

Production rules

```
LPUx x { AxcolEi } LPU
LPDx x {~AxcolEi & ~RxcolGi } LPD
LPUResRxcolo ResetRxcolo {( x & ~AxcolEi & RxcolGi ) | ~nReset} LPU
LPDResRxcolo ResetRxcolo { ~RxcolGi & nReset } LPD
```

7.11 Difference to Boahen's word serial AER

There are several differences in our AER circuits compared to Boahen's word serial AER circuits [31].

- Our pixels do not latch events, which induces more complexity in the periphery that has to handle row requests without a corresponding column request.

- In Boahen's scheme, the time window in which events of pixels in a row are collected and communicated in one burst starts with the row request of the first pixel requesting and ends with the row acknowledge. In our scheme, this time window is extended until the handshaking for communicating the row address to the off-chip receiver is completed.

- Boahen's circuit does not use a column acknowledge signal, in his scheme all active column request signals are serviced. Pixels that cross the threshold after the time window for a burst has closed, are not able to activate the column request. In our scheme, pixels that cross the threshold after the time window for a burst has closed, are still able to activate the column request. But the column state machine will not service this request in this burst. Thus active column requests are not guaranteed to be serviced. This implies that a column acknowledge signal is needed. The disadvantage of our scheme is an additional signal line and toggling of unserviced column request lines, the advantage is reduced pixel transistor count. But important is that the re-introduction of the column acknowledge (or column select as called by Boahen) does not introduce a speed disadvantage, because this signal is driven by the periphery and the driving transistors can be made large.

- For column arbitration, Boahen's scheme uses a fair arbiter just like for row arbitration. This is unnecessary, because we consider all the events transmitted in one cycle to be simultaneous, the order in which the column addresses are transmitted does not matter. Thus we use a simplified arbiter for column arbitration, which saves circuit area and speeds up the arbitration.

- We use different keeper circuits with adjustable keeping strength.

- A very minor difference is that Boahen's circuit uses separate request lines for column and row events, while we use a single request line and an additional address bit (*XSel*) distinguishing column and row requests. These two approaches do not have any practical difference.

7.12 Measurements and discussion

The new AER circuits have been used and tested in the cDVSTest20 chip fabricated in the "UMC L180 Mixed-Mode/RF" 180nm 6-metal 1-poly technology. Figures 7.21 to 7.24 show scope plots of the *nCreq*, *Rrow*, *Rcol* and the *Rcarb* signal.

Figures 7.21 and 7.22 show the chip handshaking with the CPLD of the cDVSTest PCB (see Appendix A). The hand-shaking cycle time is limited by the CPLD, it is around 90ns, allowing a peak event rate of about eleven mega-events

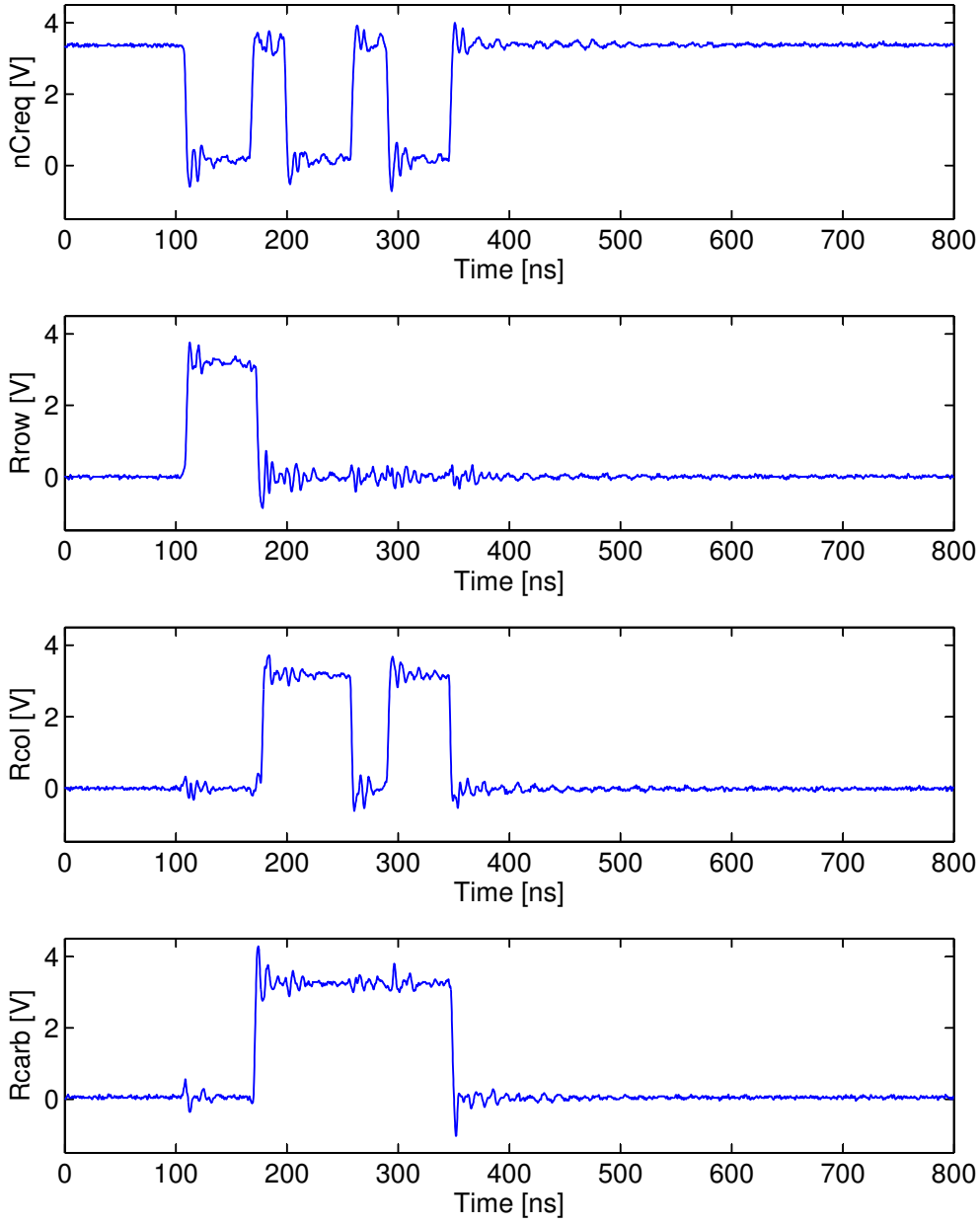


Figure 7.21: AER signals, hand-shaking with CPLD. The plots show a single burst with two column address transmissions.

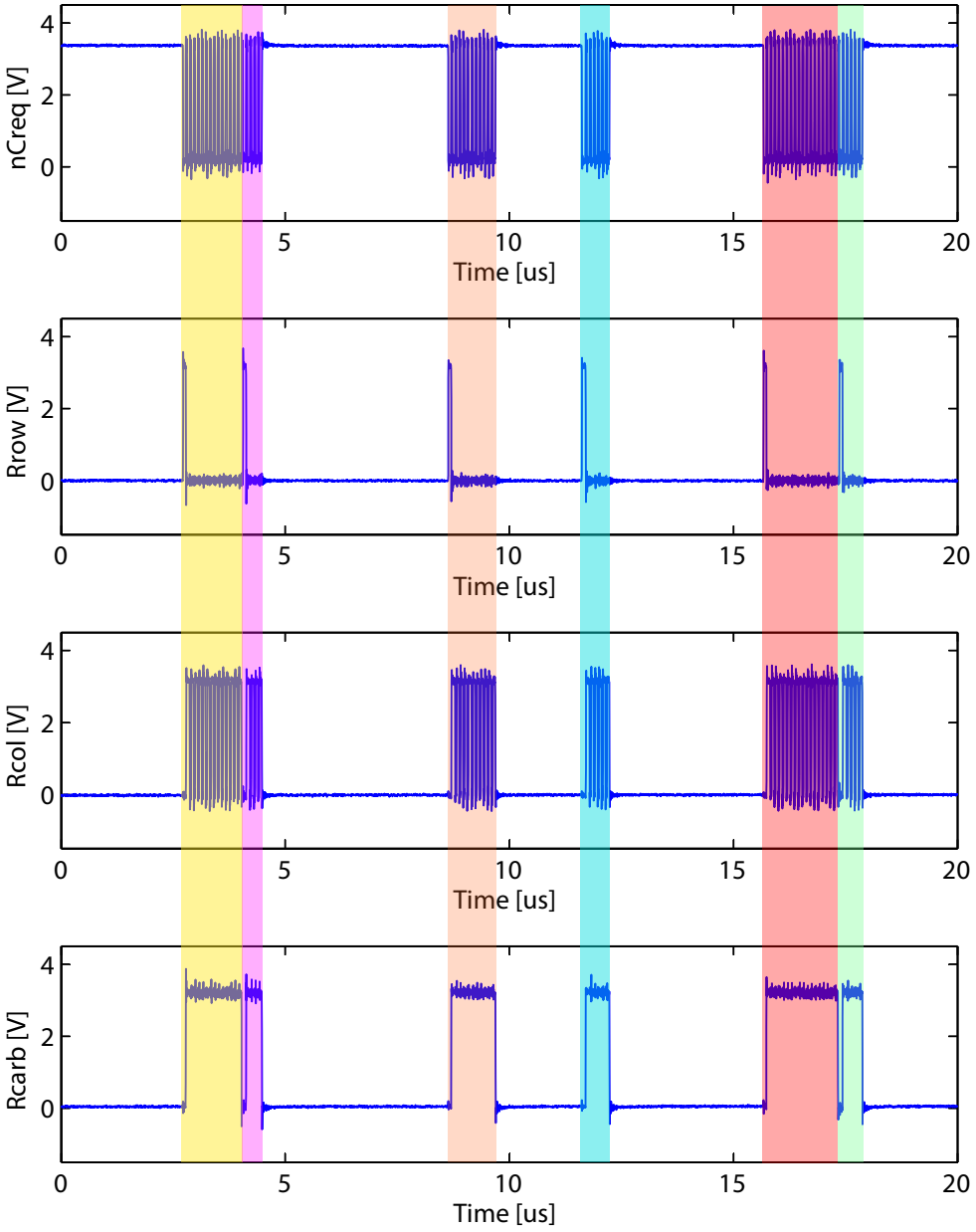


Figure 7.22: AER signals, hand-shaking with CPLD, showing six bursts of events. The bursts are marked with different colors.

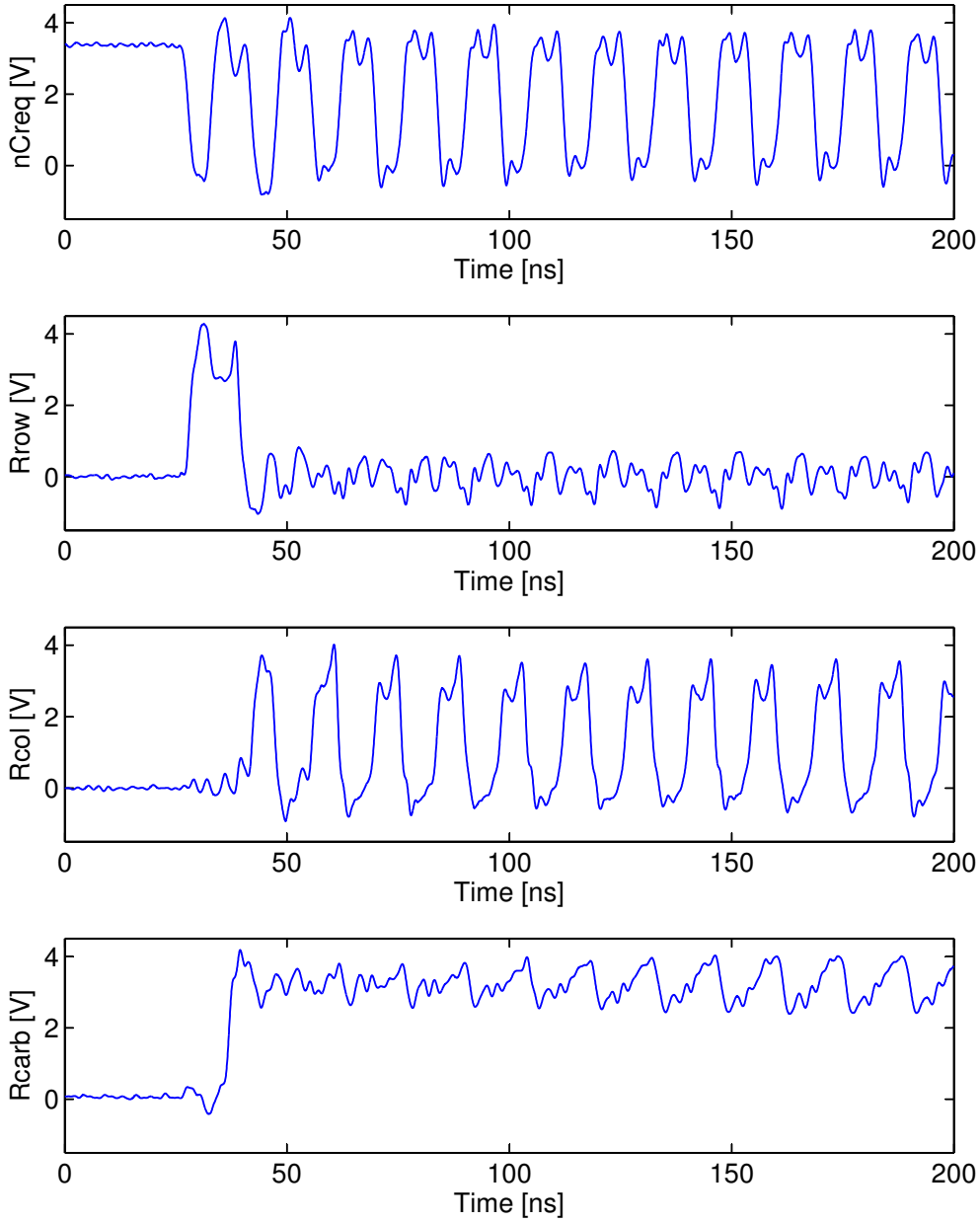


Figure 7.23: AER signals, the chip handshakes with itself, a single burst with several column addresses.

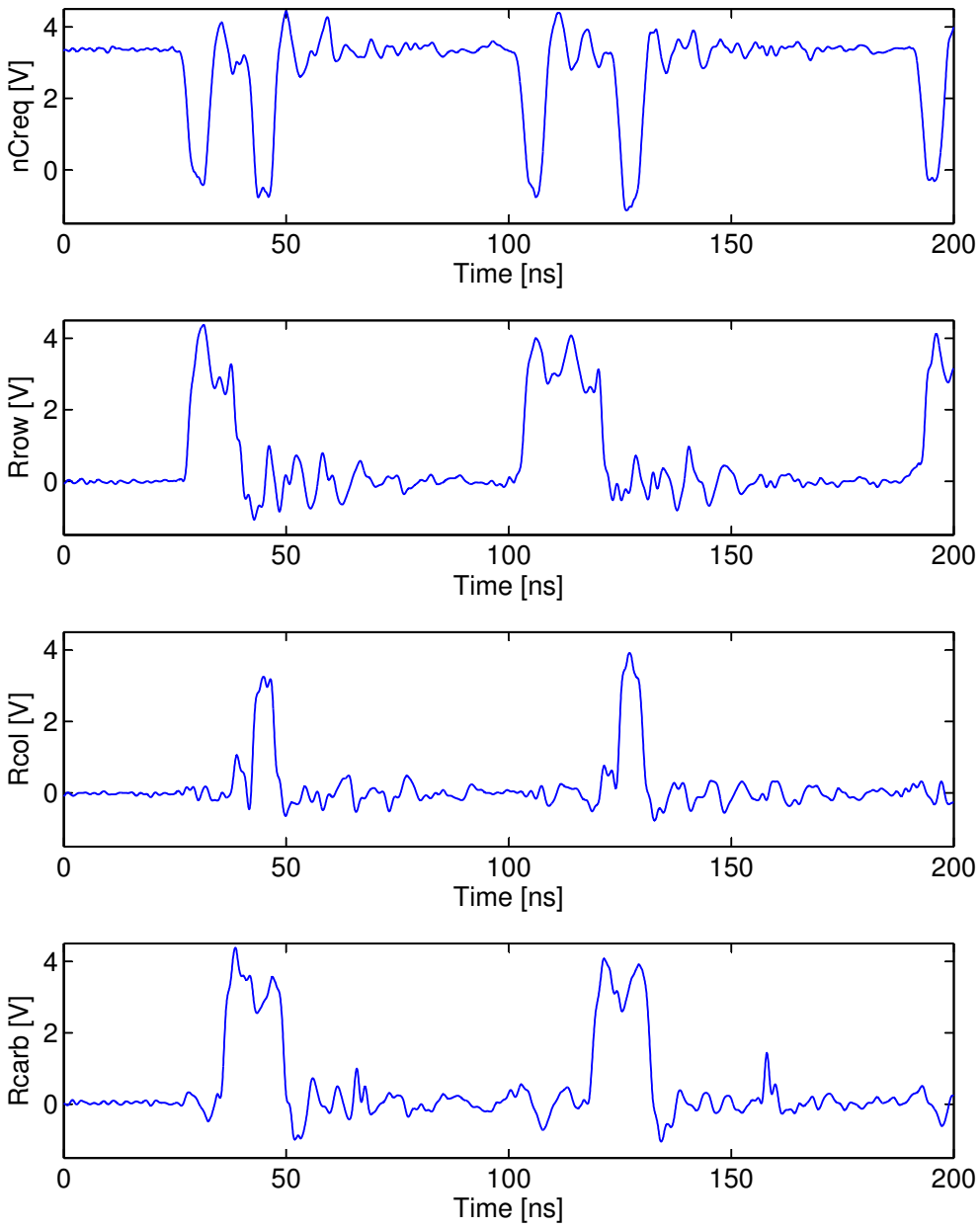


Figure 7.24: AER signals, the chip handshakes with itself, several bursts with only a single column address.

Number of column addresses	Burst duration [ns]
1	51.9
2	56.4
3	61.3
4	72
5	84.5
6	94.7
7	110
8	128

Table 7.2: Burst cycle times for word serial AER when chip request is tied to chip acknowledge. A burst consists of a row address and a number of column addresses.

per second. Fig. 7.21 shows a single burst with two column addresses, Fig. 7.22 shows six bursts with several column addresses each.

Figures 7.23 and 7.24 show the chip handshaking with itself, here the cycle time is much shorter. Fig. 7.23 shows a long burst with 12 column addresses, Fig. 7.24 shows two short bursts with only a single column address each and the beginning of a third burst.

Table 7.2 lists the burst cycle times needed to transmit bursts of different lengths. Pure handshaking cycle time is 14ns for row and for column addresses. Minimal burst duration is roughly 50ns. For more than 3 column addresses, burst duration increases roughly linearly with the number of column addresses. These times have been measured with a Agilent 16702A logic analyzer with 2ns time resolution. The cycle time compares favorably to Boahen's circuits, who reports a cycle time of 22.6ns for a column address and 72ns for cycling to a new row [33]. However, his circuits are built in 250nm process technology, our circuits are therefore expected to run faster.

However, if the chip handshakes with itself, it gets stuck after a short time, sometimes it is a few events, sometimes a few hundreds. This means that there are still implicit timing assumptions in our AER scheme, it is not delay insensitive.

Spice simulations and careful studying with the logic analyzer revealed timing assumptions that could contribute to the AER circuits getting stuck.

- Lcol might not reach Endblock. The *Lcol* pulse can be very short if the receiver is fast and *Rcarb* rises fast. So if the *Lcol* metal line has too much parasitic resistance and capacitance, at the Endblock *Lcol* might not cross the logic thresholds. Thus *AxcolE* would never be asserted and therefore the column requests not reset.

- *AxcolE* might not reach the ResetRxcol-block. If a row close to the top of the array is selected, assertion of *AxcolE* could be quite short because *RxcolE* is

taken away with *AxcolE* in each row. At the ResetRxcol-block, *AxcolE* might not cross the logic thresholds and therefore the column requests are not reset.

- The main state machine checks the down-going edge of *RxcolG* to go back to the idle state. However, the up-going transition is never checked. So if *RxcolG* goes up very slowly, and if the off-chip receiver is sufficiently fast, the main state machine might go back to idle state after transmitting all column addresses, even if *RxcolG* is still rising. This would mean that the column requests are not reset properly.

The first two problems arise because some signals may not make full transitions from GND to VDD at all positions in the chip due to parasitic resistance and capacitance. If the signals were perfectly digital (e.g. always full transitions from GND to VDD, but with arbitrary delay), these problems could not occur. Simulation with LogSpice does not reveal such problems.

7.12.1 Extension of AER scheme

In the following section we present extensions of our AER scheme that solve the timing assumptions described above. We propose to add two more signals as illustrated in Fig. 7.25:

- *LcolAck* from the Endblock back to the main state machine. The main state machine will only take *Lcol* down when *LcolAck* is active.
- Acknowledge signal from the ResetRxcol block back to Endblock to acknowledge reception of *AxcolE*.

Additionally, the main state machine should also check the up-going transition of *RxcolG*. The resulting state diagrams of the main state machine and the Endblock are illustrated in Figures 7.26 and 7.27 respectively.

In this implementation, assertion of the chip reset signal *nReset* resets all the communication circuits as well as all the pixels. This is not ideal, it would be preferable to be able to reset communication and pixels separately. This could easily be implemented if *nReset* does not assert the row and column acknowledge signals *Ayrow* and *Axcol*, instead they should be asserted with a different signal.

7.13 Conclusion

The developement of our new word serial AER scheme is a major achievement of this thesis. The scheme does not work perfectly yet, but improvements are proposed.

We do not claim that our scheme is generally better than Boahen's [31], but it is better suited for our needs by keeping transistor count in the pixel as low as possible and eliminating the bandwidth-limiting problem of the DVS128 AER circuits. Our scheme (ignoring polarity) needs 12 transistors per pixel, Boahen's

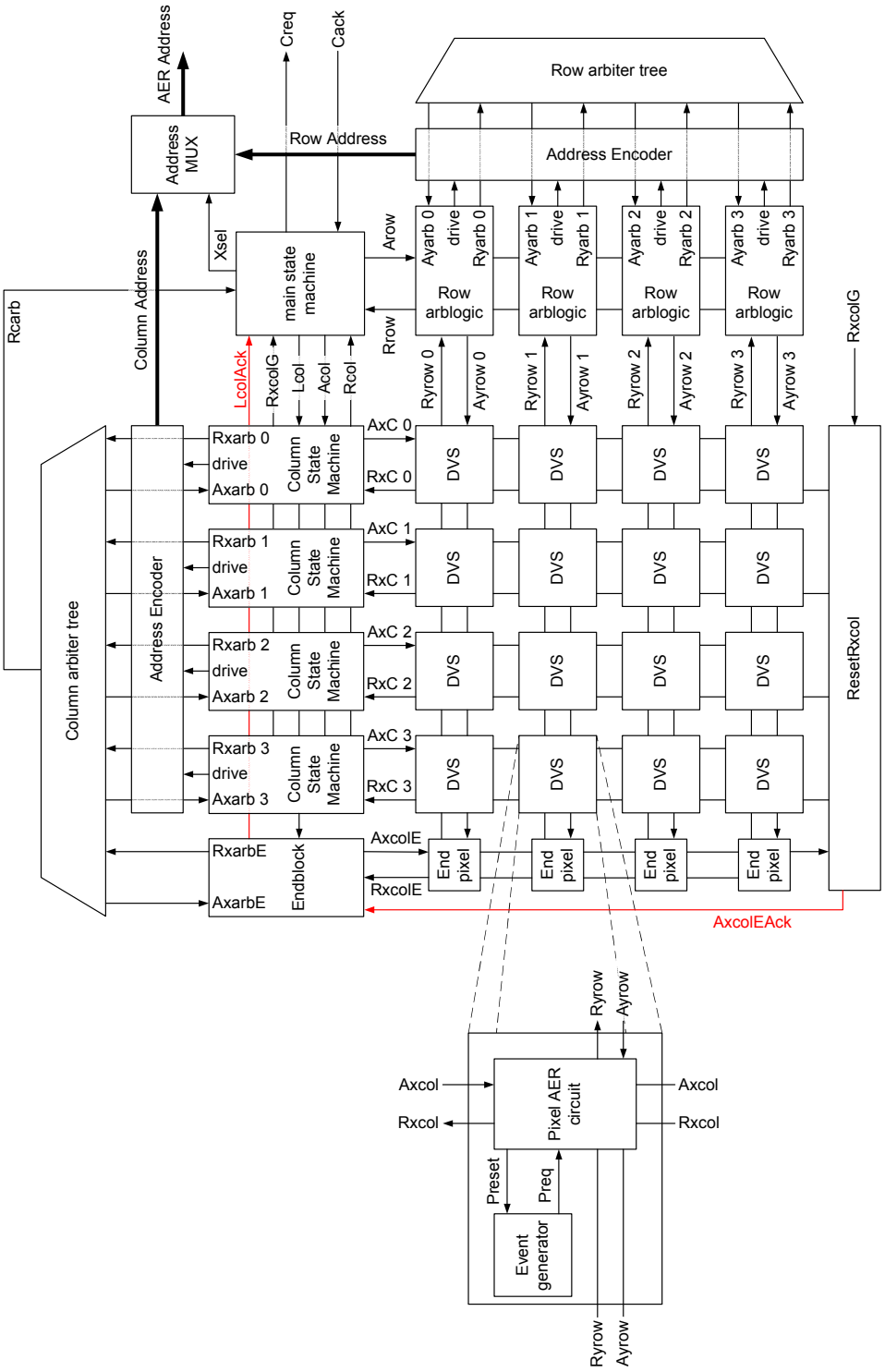


Figure 7.25: Word serial AER block diagram with timing assumption fixes marked in red.

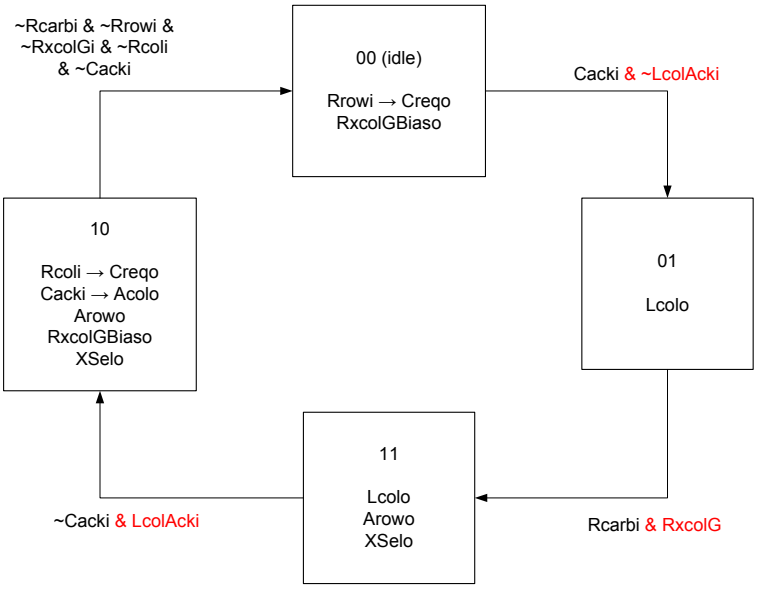


Figure 7.26: State diagram of main state machine with timing assumption fix. Changes to the originally described state machine are marked in red. These changes are a check for the up-going edge of *RxcolG* and the inclusion of the *LcolAck* input, which is checked before toggling *Lcol*.

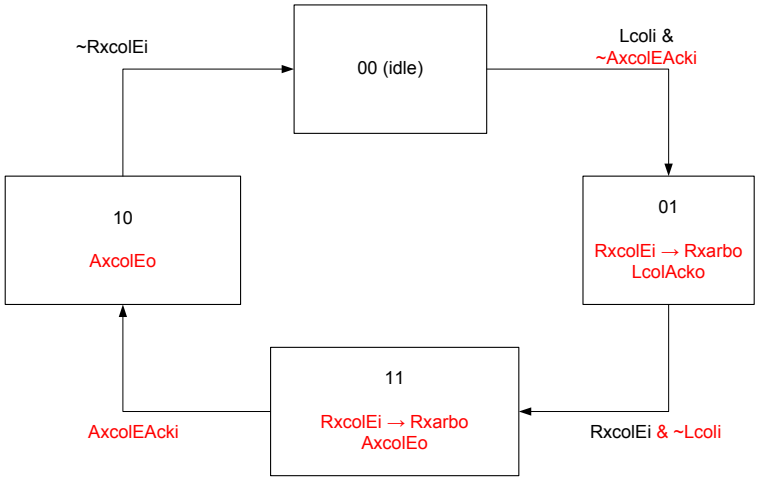


Figure 7.27: Endblock state diagram with timing assumption fix. Changes to the originally described state machine are marked in red. The changes are the inclusion of the *LcolAck* output which is used to acknowledge the reception of *Lcol*, and the *AxcolEAck* input which is checked before toggling *AxcolE*.

scheme needs 18 including the staticizers. We reduce transistor count at the expense of re-introducing the column acknowledge line.

To do without positive feedback to latch threshold crossing in each pixel increases the complexity of the whole scheme considerably and imposes the use of the Endpixels as described in section 7.7.

With these circuits, we have the means to scale up our arrays to QVGA (320×240 pixels) or even larger size without sacrificing per-pixel bandwidth and without increasing the width of the AER bus like in the ATIS [67]. The latter makes communication with the host computer less effective.

Chapter 8

cDVS pixel array with logarithmic intensity readout

8.1 Introduction

This chapter describes the final chip built during this thesis. The cDVSTest30 chip is an improved version of cDVSTest10 (see chapter 6), with fixed AER communication circuits and modified cDVS pixels. It incorporates a 32 by 32 array of logCDVS pixels which provide a dynamic color pathway, a dynamic log intensity change pathway and additionally a sustained log intensity read-out. Additionally the chip contains four 32 by 32 arrays of variants of the DVS pixel. It uses the word-serial AER circuits described in chapter 7.

Section 8.2 describes the dynamic pathways of the logCDVS pixel, which are only slightly modified from the cDVS pixel. Section 8.3 illustrate how the logarithmic readout is implemented and section 8.4 shows the architecture of the cDVSTest30 chip. Section 8.5 provides measurement results.

8.2 logCDVS pixel dynamic pathways

The pixel is based on the pixel presented in chapter 6, but uses only a single stage amplifier for the color pathway. The modified color change pathway is shown in Fig. 8.1, the input capacitances of the summing amplifier are 2% different as suggested by the measurement in Fig. 6.7. The intensity change pathway is identical to the pixel presented in section 6.1.2.

8.3 Log intensity readout

The pixel includes a common gate transistor in the front-end which converts the total current of the two photodiodes logarithmically to a voltage. The pixel can be

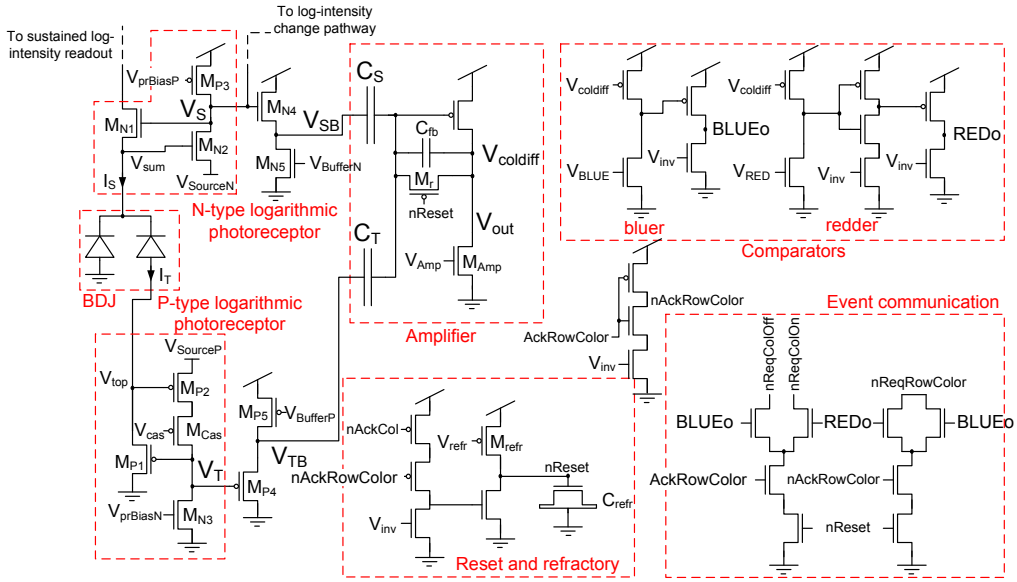


Figure 8.1: cDVSTest30 color change pathway

addressed and the voltage at the source of the read-out transistor – and therefore the light intensity – be read out.

Unlike the ATIS [67] produced by the Austrian Institute of Technology, the readout of the intensity is completely decoupled from the event pathway. In the ATIS, an intensity reading is initiated with every spike of the DVS pixel. This is optimal if only the output of the intensity pathway is considered, as each time a change occurs, a new intensity value is transmitted. However, if one also considers the information contained in the DVS spikes, this strategy is not optimal anymore. The DVS spikes already contain the information needed to calculate the new intensity value, as a spike from one pixel always has the same meaning.

Because there are background and noise spikes, the refractory period as well as mismatch in the spiking thresholds, it is not enough to just record an image at start-up and then update it with the DVS spikes. Therefore, we propose to read the intensity values with a slow frame rate (a few frames per second), and update the intensity values with the DVS spikes in between frames. This strategy is more efficient than the ATIS for high activity and less efficient for low activity. One can also imagine more sophisticated read-out schemes where the frame rate is dependent on the event rate to increase efficiency for low activity.

The ATIS implementation is inefficient in terms of silicon area and quantum efficiency because it adds a second photodiode just for intensity reading and has a high pixel complexity. The implementation presented here uses the same photodiode for DVS and intensity and adds just seven transistors, three of them analog, only marginally increasing the pixel size. With our implementation we can fit the color and intensity change DVS pathways plus intensity read-out into the same area

as the ATIS pixel, which provides no color information at all.

8.3.1 Pixel circuit

Logarithmic current mode image sensors have been presented long ago [130], the circuit our pixel is based on was presented in 1992 [131]. The basic idea is that the photocurrent is converted logarithmically to a voltage by means of a transistor. This kind of pixel is time-continuous and has a very high dynamic range due to the logarithmic compression. But these kind of image sensors have three intrinsic problems:

1. Threshold mismatch of the diode connected transistor, resulting in a high fixed pattern noise (FPN).

2. Low voltage swing, around 400mV for six decades of illumination, which limits signal-to-noise ratio (SNR).

3. Long settling time for low photo-currents after a pixel has been selected for readout. This is due to charge injection and capacitive coupling to the readout node which is a high impedance node for low photo-currents.

Several methods of FPN correction for current mode logarithmic sensors have been presented, the simplest being recording of the response to a uniform scene, which is later used for correction. Other approaches try to alter the pixel response to become more uniform, for example using feedback circuits [132] or floating gates [133, 134]. These methods however require substantial pixel area. Ni and Matou presented a correlated double sampling scheme which uses the photodiode in photovoltaic mode [135], while Lai et al. combined the correlated double sampling scheme with lateral bipolar current amplification [136]. These double sampling schemes have the disadvantage that the calibration is only as accurate as the leakage current matching between transistors, which is poor because the transistors work in deep subthreshold.

In 2000, Kavadias et al presented a compact calibration scheme [137], which was later extended and improved by Steve Collins' group [138, 139]. We follow here the original idea of Kavadias which only corrects offset mismatch. This correction is not as effective as the scheme proposed by Collins, but it is possible to do the correction entirely on chip before analog-to-digital conversion. The idea is to read the voltage of the source of the readout transistor when it supplies first the photocurrent and afterwards a single common (or column parallel well matched) reference current. Taking the difference of these two readings removes the effect of threshold voltage mismatch in the readout transistors. This method can be applied to the logCDVS pixel as well.

Fig. 8.2 shows a logarithmic current mode pixel with a reference current source I_{ref} for mismatch reduction. By doing two readings, one can get rid of offset mismatch. Adding a third reading (two different reference currents) allows calibration for gain mismatch.

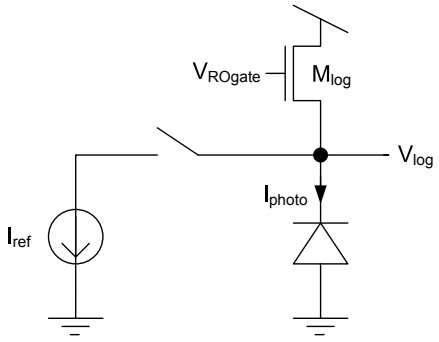


Figure 8.2: Logarithmic current mode pixel with reference current source.

The voltage at the source of the readout transistor $M_{\log}$ is

$$V_{\text{logPhoto}} = \kappa V_{ROgate} - U_T \ln \frac{I_{\text{photo}}}{I_0}. \tag{8.1}$$

When the reference current is added

$$V_{\text{logRef}} = \kappa V_{ROgate} - U_T \ln \frac{I_{\text{photo}} + I_{\text{ref}}}{I_0}. \tag{8.2}$$

The difference of these two readings is

$$V_{out} = U_T \ln \frac{I_{\text{photo}} + I_{\text{ref}}}{I_0} - U_T \ln \frac{I_{\text{photo}}}{I_0} \tag{8.3}$$

$$= U_T \ln \left(\frac{I_0}{I_{\text{photo}}} \frac{I_{\text{photo}} + I_{\text{ref}}}{I_0} \right) \tag{8.4}$$

$$= U_T \ln \frac{I_{\text{photo}} + I_{\text{ref}}}{I_{\text{photo}}}, \tag{8.5}$$

which, for $I_{ref} \gg I_{\text{photo}}$ simplifies to

$$V_{out} = U_T \ln \frac{I_{ref}}{I_{\text{photo}}}. \tag{8.6}$$

eliminating the dependency of the output voltage on the off-current of $M_{\log}$. If the reference current is the same for all the pixels, offset mismatch is removed. Additionally, this double sampling also removes offset mismatch introduced by the readout buffer. The reference current has to be chosen so that the readout transistor $M_{\log}$ stays in subthreshold.

Ideally, if for the reference reading only the reference current was used and the photocurrent switched to somewhere else, the condition $I_{ref} \gg I_{sum}$ would not be needed. The reference current could be chosen somewhere in the middle of the photocurrent range, which would improve the calibration. We don't use this in our

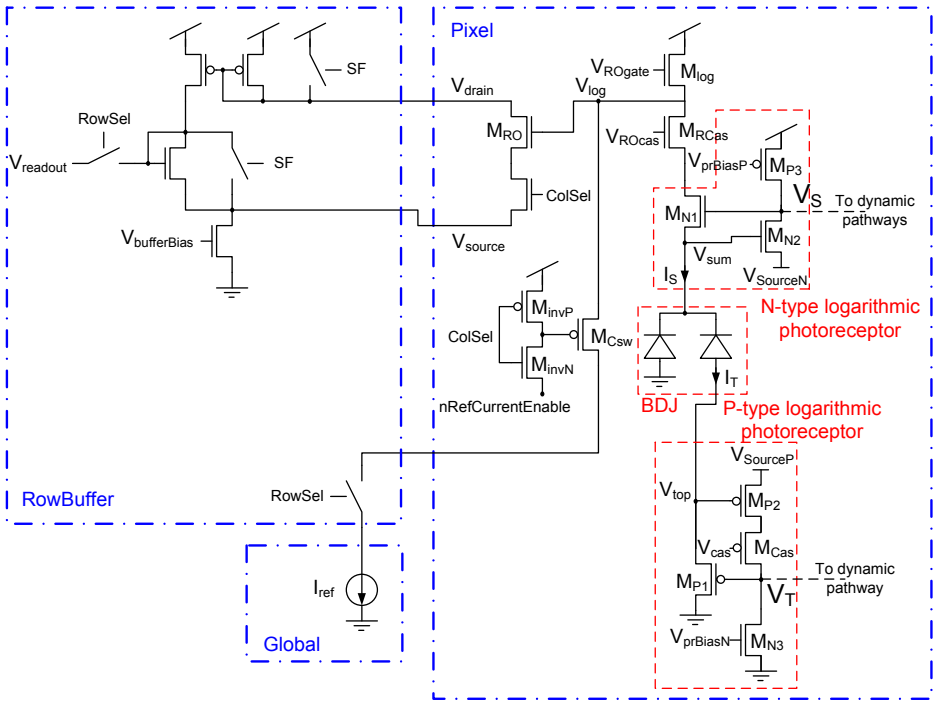


Figure 8.3: logCDVS pixel front-end with logarithmic photocurrent read-out and row read-out buffer.

pixel because of the added complexity and because we want to switch as little as possible to avoid coupling to the event generation pathway.

Even though various methods of FPN correction have been proposed, logarithmic current mode image sensors have mostly been abandoned in recent years in favor of integration mode APS circuits using a stepped reset voltage to achieve a logarithmic transfer function [45, 140], which improve image quality due to higher signal swing and shorter settling times. For us long settling times for low photocurrents pose less problems because we do not need fast frame rates.

Low voltage swing can to some extent be improved by on-chip amplification, but of course having a higher voltage swing at the sensor front-end is better. Here we compromise on SNR to get a compact and simple implementation of logarithmic compression instead of using more complex integration-mode schemes with stepped reset [45, 140].

Fig. 8.3 shows the complete logCDVS pixel front-end with logarithmic current readout and also the row buffer and the global reference current source. We added the cascode transistor M_{RCas} between the current-to-voltage converter transistor M_{log} and the feedback transistor M_{N1} to shield the event generation pathway from switching noise induced from addressing the pixel and enabling the reference current.

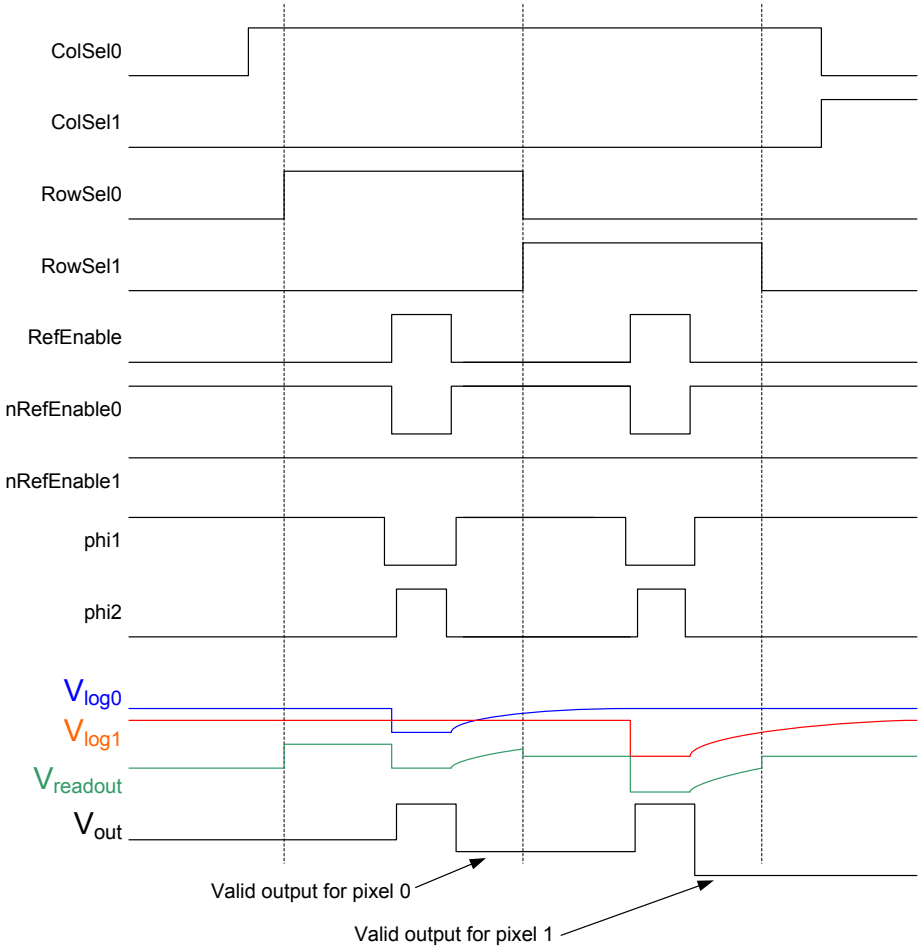


Figure 8.4: Signals for log intensity readout

The readout transistor M_{RO} is enabled by the column select transistor and connected to the row buffer. Fig. 8.4 shows the signal sequence for selecting a pixel.

Transistors M_{invN} and M_{invP} perform an NAND operation between the signals *ColSel* and *RefCurrentEnable* to steer the reference current to the currently selected pixel (if both *RowSel* and *ColSel* are active), if the signal *nRefCurrentEnable* is low. However, after fabrication of this chip, we realized that this NAND-operation in the pixel is completely redundant, because this operation is already done in the periphery as indicated in Fig. 8.5.

8.3.2 Peripheral readout circuits

The peripheral circuits consist of row buffers, row and column scanners to address the pixel, a difference amplifier and a non-overlapping clock generator, which are all explained in the following subsections. The control signals needed to control the scanners and the difference amplifier are supplied externally by a CPLD (see appendix A). The chip relies on an external analog-to-digital converter.

Row buffer

Fig. 8.3 shows the logCDVS pixel front-end and the row read-out buffer "Row-Buffer". It is a 5-transistor voltage buffer which can also be configured as source-follower buffer by means of two transmission gates (a full transmission gate between V_{source} and V_{readout} and a pFET at V_{drain}) which are controlled by the signal *SF*, allowing to compare these two buffer variants.

Pixel addressing circuits

The pixels are addressed using shift-register scanners, illustrated in Fig. 8.5. The same scanners are also used to steer the single reference current source to the addressed pixel. The implementation of a bigger array would probably need a reference current per row to increase readout speed, but introducing mismatch in the reference current sources which has to be corrected [138]. The row-select scanner selects the output of a row buffer and connects it to the difference amplifier. The on-chip calibration can also be switched off and V_{readout} is then monitored directly.

This design is rotated compared to most image sensors, which have an array of column buffers at the bottom of the pixel array. The arrays are then scanned row wise, our array is scanned column wise. This was done because there was more space available to the left of the array to implement the row buffer circuits.

Difference Amplifier

A single switched-cap differencing amplifier (Fig. 8.6) amplifies the difference between the photocurrent reading and the reference current reading. It works in three phases:

1. *Phi1* is active: The photocurrent reading is sampled on the gate of M_{buf}.

2. *Phi2* is active: The reference current is enabled and the voltages corresponding to the reference current and the sampled photocurrent reading are applied to the plates of the capacitor C_{in}. Thus the voltage over C_{in} equals the difference between the photocurrent reading and the reference current reading. Capacitors C_1 and C_2 are reset.

3. *Phi1* is active again: C_{in} is connected to the capacitive amplifier. The charge on C_{in} splits between C_{in} and C_1. The voltage over C_1 is amplified by a factor

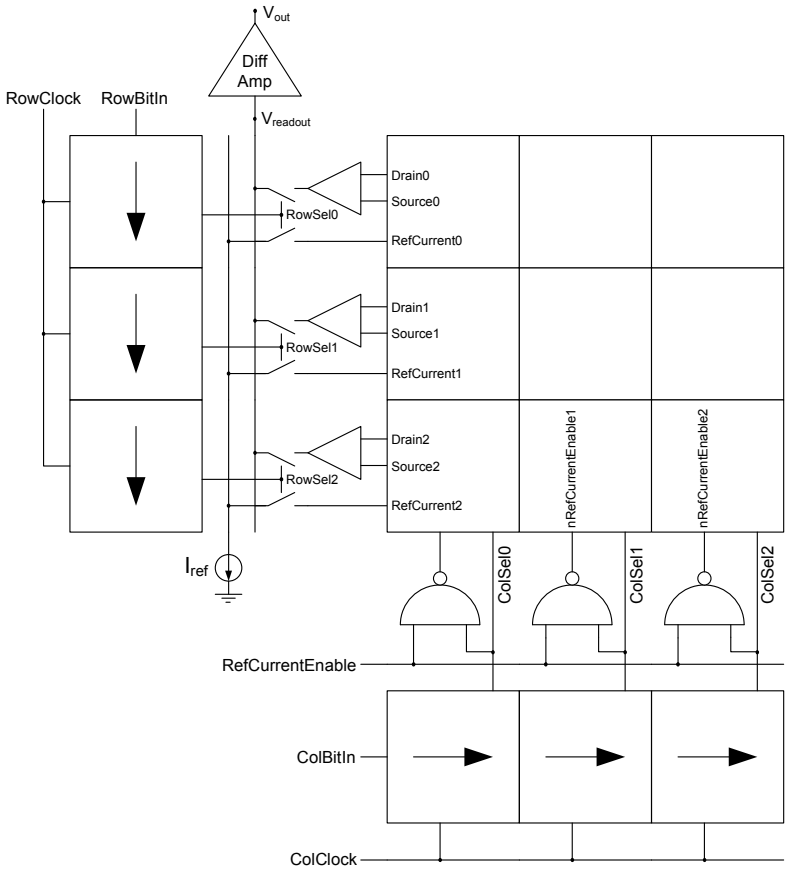


Figure 8.5: Pixel addressing and reference current steering.

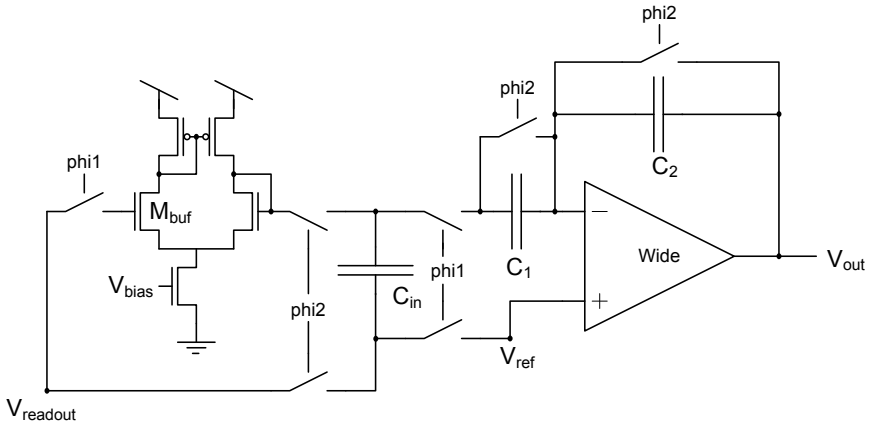


Figure 8.6: Difference amplifier for log-intensity readout. All the switches are implemented as full transmission gates, *Phi1* and *Phi2* are non-overlapping clocks. $C_{in} = 570fF, C_1 = 380fF, C_2 = 114fF$, which provides a total gain of 2.

of C_1/C_2 and the output V_{out} is valid. This third phase can already be the first phase for the next pixel.

The total gain of this amplifier is

$$A = \frac{C_{in}}{C_{in} + C_1} \frac{C_1}{C_2}. \tag{8.7}$$

For future revisions, more sophisticated amplifier topologies can be used, for example as presented by Johns and Martin, page 432 [123].

Non-overlapping clock generator

Two cross-coupled NOR gates with delay create the *Phi1* and *Phi2* clocks for the difference amplifier from the single phase input clock *RefCurrentEnable*. Fig. 8.7 shows the schematics. The non-overlap is determined by the inverter delay τ.

Signal sequence

Fig. 8.4 shows time-lines of all the signals relevant for the log intensity readout for two different pixels in the same column. First pixel 0 is selected (*ColSel0* and *RowSel0* high) and the voltage corresponding to the photocurrent is read and sampled on the gate of M_{buf} (Fig. 8.6). Then the reference current is enabled and the corresponding voltage is read. When *RefCurrentEnable* goes low again, the voltage corresponding to the photocurrent and the voltage corresponding to the reference current are applied to the difference amplifier, the output voltage V_{out} is valid for pixel 0. Then the next pixel is selected by clocking the row shift register, *RowSel0* goes low and *RowSel1* goes high.

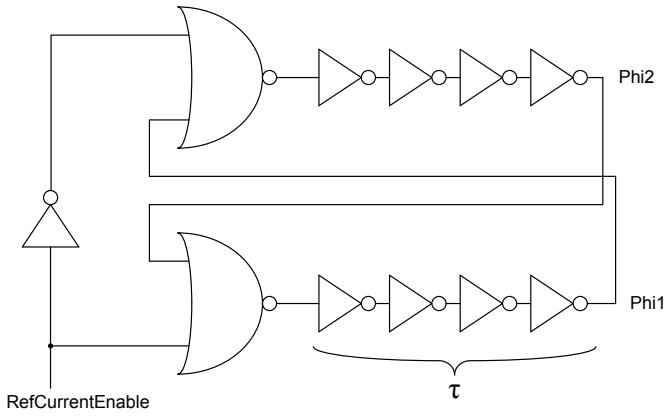


Figure 8.7: Non-overlapping clock generator

8.4 cDVSTest30 chip

The cDVSTest30 chip is built in the "UMC L180 Mixed-Mode/RF" 180nm 6-metal 1-poly process with metal-insulator-metal (MIM) capacitors using two Europractice miniASIC slots [99]. Die size is $3.24 \times 1.525 mm^2$. Table 8.1 list the specifications of this chip focusing on the logCDVS pixel array. The power consumption of the analog part of the chip is relatively high, it is dominated by the photoreceptor bias current, which we needed to set to more than 100nA for correct functioning of the pixel.

The complete chip architecture is shown in Fig. 8.8. A die photo is shown in Fig. 8.9. As in Lichsteiner's work [39], the layout of the logCDVS pixel is drawn for use in a quad-mirrored configuration to separate analog and digital parts of the pixel and share bias and power lines between pixels. Fig. 8.10 shows the layout of 2×2 logCDVS pixels.

8.4.1 AER

The AER circuits of cDVSTest30 contain slight modifications compared to the circuits described in chapter 7, although not yet the improvements proposed in section 7.12. The modifications in cDVSTest30 do not improve the performance, on the contrary they necessitate slower handshaking with the CPLD to prevent getting hung in a dead-locked state. Thus we do not discuss these modifications here.

Because we suspected problems with the keeper circuits for the row and column request signals, we added an additional safety measure which allows us to switch on static pull-ups on the column and row request signals. These proved to be unnecessary though.

Process	UMC 180 nm, N-Well, 6 Metal, 1 Poly, MIM-caps
Functionality	Dynamic vision sensor test chip with log-intensity change, color change and sustained log-intensity pathway
Array size	32×32
Pixel size	$29\mu\text{m} \times 29\mu\text{m}$
Photodiode Area	$144\mu\text{m}^2$
Fill factor	17%
Number of elements per pixel	84 transistors, 5 capacitors 2 MOS capacitors
Designed Amplifier Gain $\frac{V_{\text{diff}}}{V_{SBi}}, \frac{V_{\text{coldiff}}}{V_{SB}}, \frac{V_{\text{coldiff}}}{V_{TB}}$	26 dB, 27 dB, 27 dB
Power supply	Core 1.8V, digital I/O 3.3V
Current consumption	3.3V I/O: 0.8mA @1.8Meps, 0.4mA @20keps 1.8V digital: 0.9mA @1.8Meps, 0.2mA @20keps 1.8V analog: 17.8mA

Table 8.1: cDVSTest30 specifications.

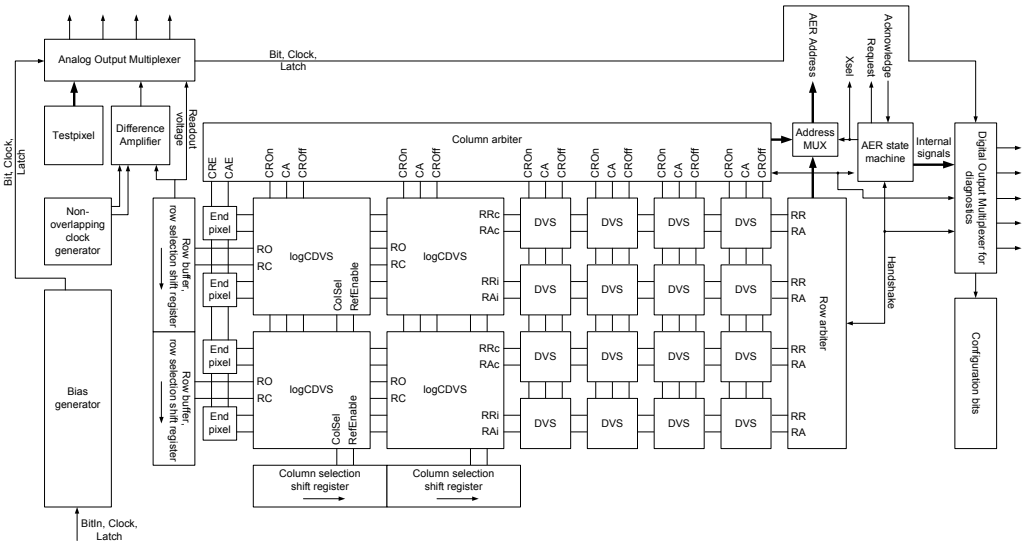


Figure 8.8: cDVSTest30 chip architecture illustrated with reduced array size. Real array sizes are 32×32 for the logCDVS pixels and 64×64 for the DVS pixels. The bias generator, the analog and the digital output multiplexer and the configuration bits are controlled with a single shift register. The signals are: RO: intentisty readout voltage, RC: reference current, RRc and RAc: row handshake signals for color change event, RRi and RAi: row handshake signals for intensity change event, XSel: indicates row or column address.

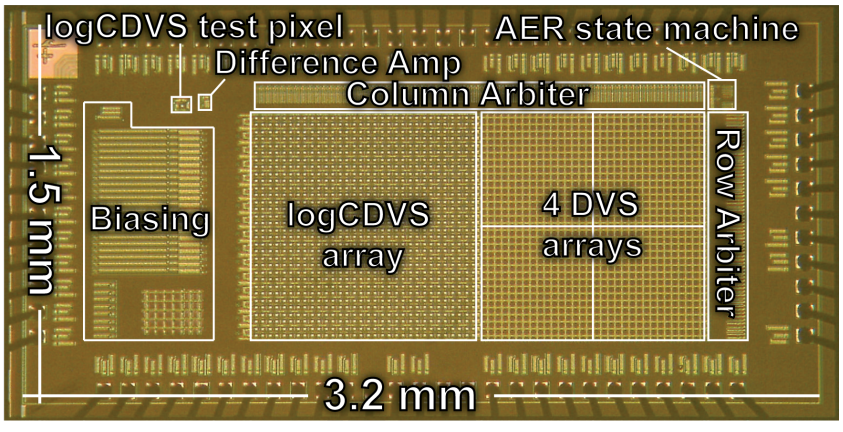


Figure 8.9: cDVSTest30 die photo.

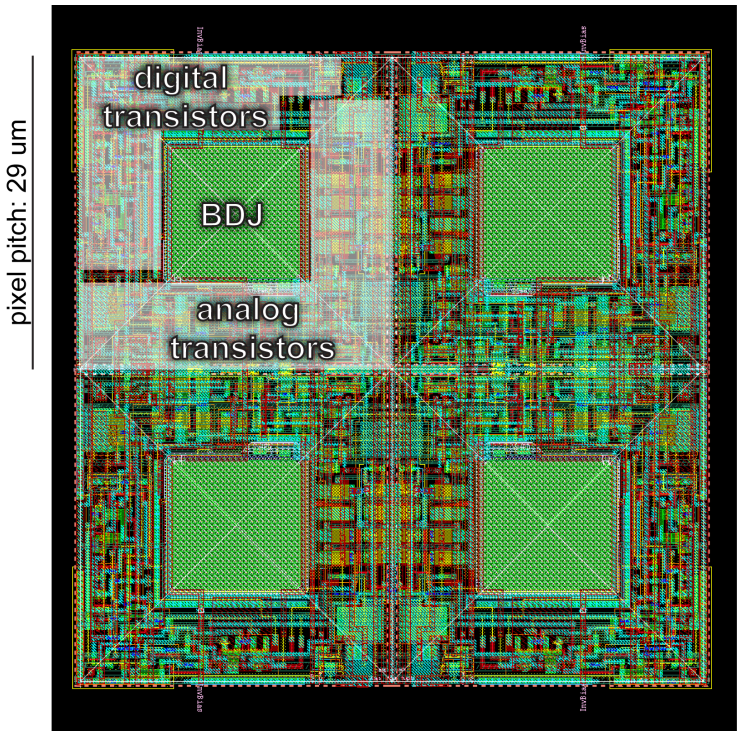


Figure 8.10: Layout of 2 by 2 logCDVS pixels

8.4.2 Bias generator

The cDVSTest30 chip includes the same bias generator as the cDVSTest10 chip, with the addition of a simple voltage generation circuit. It is used for generating the reference voltage for the difference amplifier used for log readout (section 8.3). It is a current source that uses an external resistor to generate a voltage.

8.4.3 DVS pixel variants

The chip also includes a 64×64 array of modified DVS pixels designed by Patrick Lichtsteiner. There are 4 different pixel variants of 32×32 pixels each to test different possible circuit improvements. These pixels are not part of this thesis and therefore not described here. These DVS pixels have a $14.5\mu m$ pitch and therefore occupy only one fourth of the area of a logCDVS pixel.

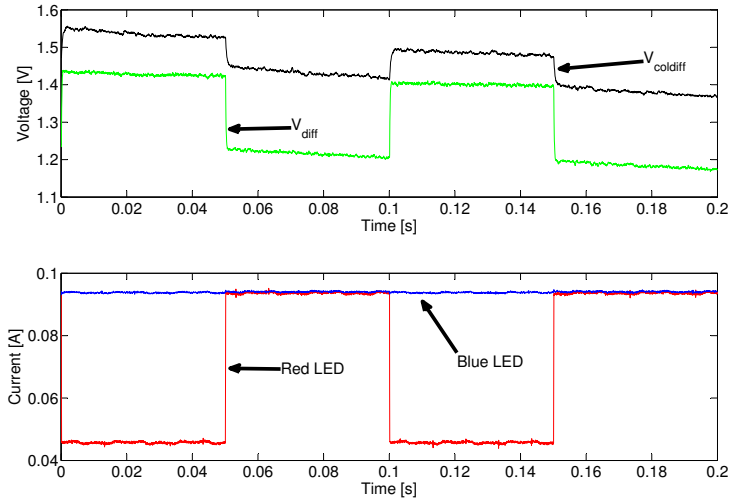
8.5 Measurements

8.5.1 Test-pixel

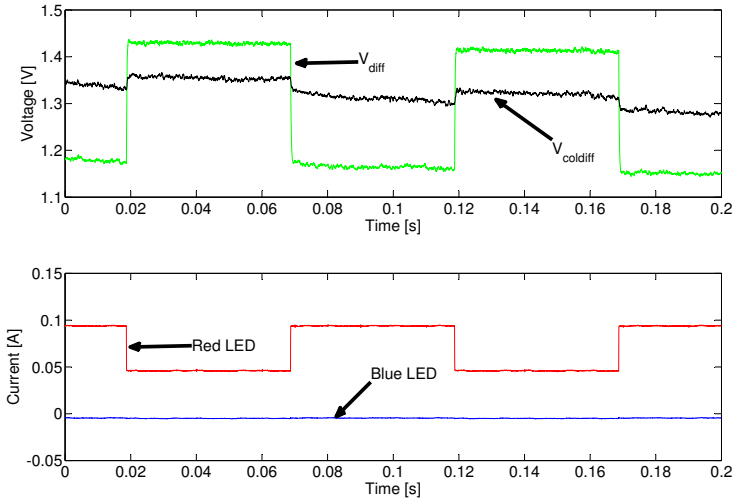
Here we provide measurement results of the test pixel. Fig. 8.11 shows that the color pathway responds much more to color change than to intensity change. The response to intensity change is reduced significantly compared to the cDVSTest10 test pixel (Fig. 6.9), but of course also the response to color change is reduced due to the lower gain. Important is that the ratio of the response to color change and the response to intensity change improved from around 2.5 in cDVSTest10 to 4 in cDVSTest30 for these input stimuli.

Fig. 8.12 and 8.13 show the response to similar inputs but attenuated by a factor 100 (ND2) and 1000 (ND3) respectively. The V_{diff} and V_{coldiff} traces become significantly noisier. For the ND3 trace, the V_{coldiff} trace shows a bandpass-like response due to different time constants for V_S and V_T in the front-end. This can partially be overcome by lowering the bias current in the differentiator stage (lowering V_{amp}, see Fig. 8.1), which also lowers the noise level at the expense of a slower BLUER comparator. Fig. 8.14 illustrates the response to intensity change with the V_{amp} bias turned to the lowest possible level. Noise level and bandpass-behavior are significantly reduced. The cascode transistor M_{cas} did not improve the bandpass-like behavior.

In Fig. 8.11 we can observe that the V_{diff} and V_{coldiff} traces are strongly tilted, and the tilt is reduced to almost zero in Fig. 8.12 to 8.14. This means that for strong illumination, the floating nodes of the capacitive amplifiers are leaking to V_{DD} due to parasitic photocurrents in the active-to-well diode of the reset switches M_r (Fig. 8.1) and M_{ri} (Fig. 6.5), even though these transistors are completely covered with metal layers. Fig. 8.15 illustrates the leaking rate of the test-pixel in different illumination. The leakage could possibly arise because the n-well these switches are sitting in is not completely covered by metal. Even though there are other transistors between M_r and M_{ri} and the holes in the metal cover, carriers

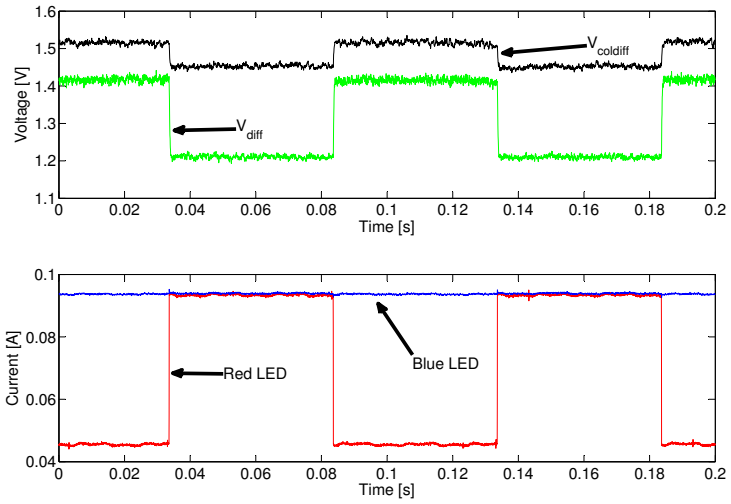


(a) Response to color and intensity change. The intensity changes from $10.7W/m^2$ to $7.38W/m^2$, the mean wavelength changes roughly from 556nm to 526nm.

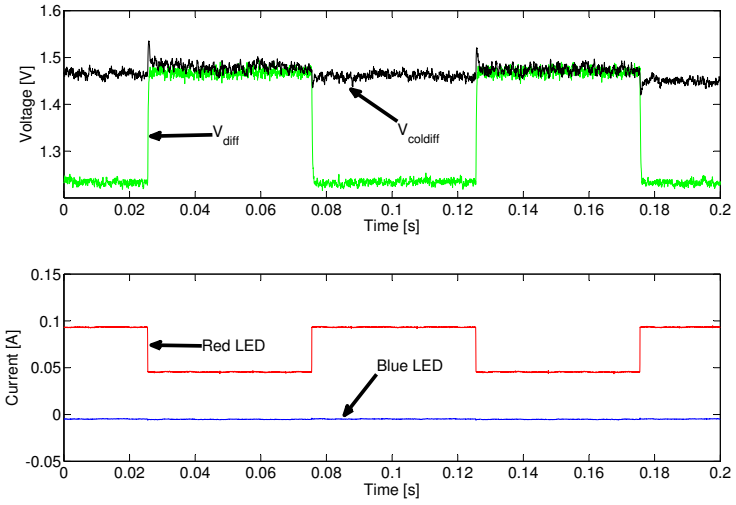


(b) Response to intensity change of roughly one octave ($6.82W/m^2$ to $3.34W/m^2$). The blue LED is turned off, the chip is only illuminated by the red LED, thus there is no color change.

Figure 8.11: Response of cDVS pixel to square wave input with thresholds set so that no events are produced.

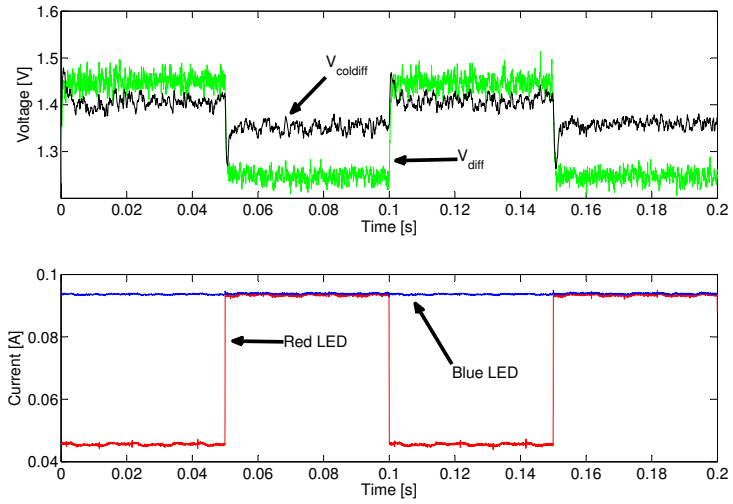


(a) Response to color and intensity change.

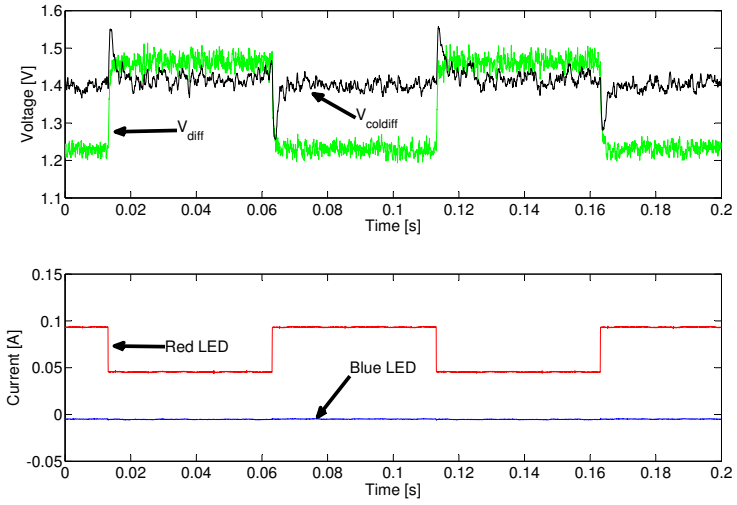


(b) Response to intensity change of one octave. The blue LED is turned off, the chip is only illuminated by the red LED, thus there is no color change.

Figure 8.12: Response of cDVS pixel to square wave input with thresholds set so that no events are produced, light attenuated with ND2 filter.



(a) Response to color and intensity change.



(b) Response to intensity change of one octave. The blue LED is turned off, the chip is only illuminated by the red LED, thus there is no color change.

Figure 8.13: Response of cDVS pixel to square wave input with thresholds set so that no events are produced, light attenuated with ND3 filter.

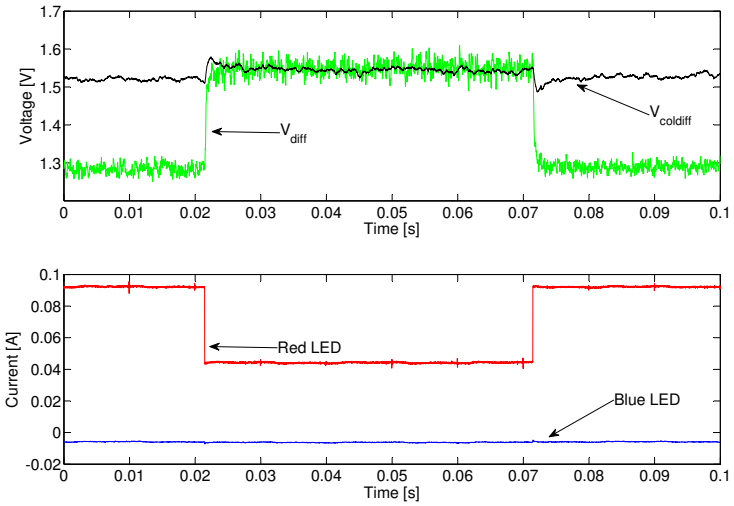


Figure 8.14: Response of cDVS pixel to intensity change of one octave with thresholds set so that no events are produced, the light is attenuated with a ND3 filter. For this experiment a lower amplifier bias was used to decrease bandwidth. The blue LED is turned off, the chip is only illuminated by the red LED, thus there is no color change.

generated in the parasitic well-to-substrate photodiode might diffuse to the switches and create parasitic currents. The switches should thus be better protected by well contacts to suck up photo-generated minority carriers. Table 8.2 shows quantitative measurements of the background for V_{coldiff} and V_{diff}. The two nodes do not respond equal to red and to blue light, which is probably due to different placement and different surroundings of the switch transistors.

To illustrate color sensitivity, Fig. 8.16 shows the response to a color sweep from red to blue and vice versa. Comparing to Fig. 6.11, we see that with a single stage amplifier, the problem of charge injection is much less pronounced and we can set the thresholds much closer to the balance level. However, there is still charge injection and thus we can not set the BLUER threshold as close to the settling voltage as the REDDER threshold, which explains part of the asymmetry in the number of REDDER events compared to the number of BLUER events. Additional asymmetry results because of the large leakage current due to parasitic photocurrents at high illumination, as explained above. With almost 20dB less gain compared to the pixel presented in Chapter 6, we get almost as many events for a full sweep from blue to red (11 compared to 12), but only five spikes for the sweep from red to blue. With lower illumination (Fig. 8.17) the leakage is lower and the response is more symmetric with six BLUER and nine REDDER events. Even though in this experiment the intensity stays roughly constant, the intensity change detection pathway responds with two ON and two OFF events per cycle,

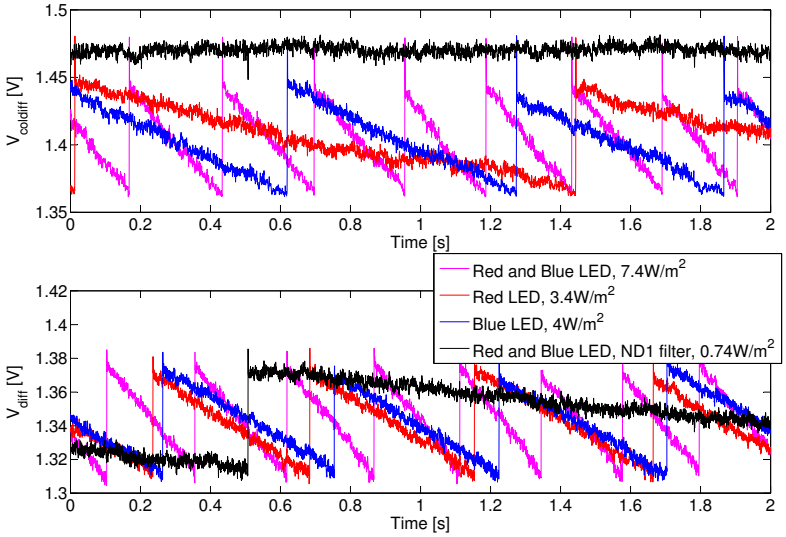


Figure 8.15: Leakage due to parasitic photocurrent in the test-pixel for constant illumination at different light intensities.

Irradiance $[W/m^2]$		Background rate [Hz]	
		V_{coldiff}	V_{diff}
Red LED	13.2	19	6.72
	6.68	9.5	3.37
	3.32	4.375	1.66
	1.24	1.7	0.625
	0.73	0.85	0.35
	0.15	<0.1	<0.1
Blue LED	7.3	5.85	3.73
	4.09	3.01	2.26
	2.14	1.23	1.325
	0.84	0.32	0.85
	0.38	<0.1	0.325
	0.05	<0.1	<0.1

Table 8.2: Background event rate for varying illumination.

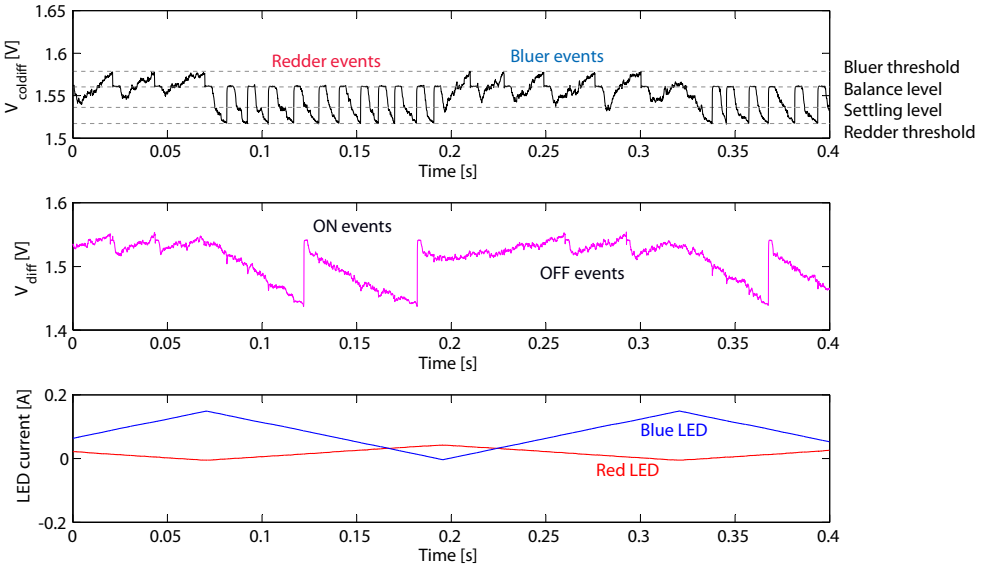


Figure 8.16: Testpixel color sensitivity

because the spectral sensitivity of the summed BDJ currents is not flat.

Fig. 8.17, 8.18 and 8.19 show the color sweep response when the light is attenuated with Kodak Wratten neutral density filters. The ND1 curve (factor 10 attenuation) shows a more symmetric response compared to Fig. 8.16, while in the ND2 curve (factor 100 attenuation, Fig. 8.18) the response is clearly noisier and the BLUER spikes have high jitter. The ND3 curve (factor 1000 attenuation, Fig. 8.19) shows still a nice response of the intensity pathway, but the response of the color pathway is very noisy.

At low photocurrents, an additional problem arises, as illustrated in Fig. 8.20. The intensity pathway couples strongly to the photoreceptor voltages V_S and V_T, which then causes a false event in the color pathway. The coupling could not be affected by the source follower bias V_{BufferN}. At low intensities, the photoreceptors need a significant amount of time to recover from the coupling, thus generating a new event right away if the refractory time is shorter than the recovery time. At high illumination, the peak in V_T is sharp enough to be completely filtered due to the limited bandwidth of the differentiator.

A closer look at the pixel layout reveals that the reset node of the log intensity change detection pathway *nReseti* runs along the edge of the photodiode and thus capacitively couples to it (see Fig. 8.21). Adding a small capacitor of 0.1fF between *nReseti* and V_{top} enables us to recreate the effect of the coupling in simulation. The layout can easily be improved to reduce this coupling and a chip including a testpixel with improved layout has already been taped out.

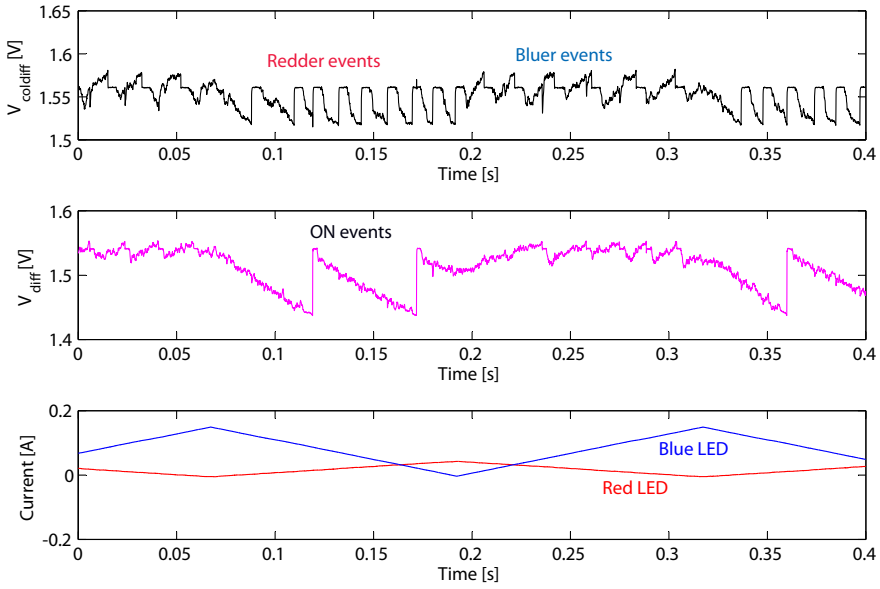


Figure 8.17: Test-pixel color sensitivity with ND1 filter

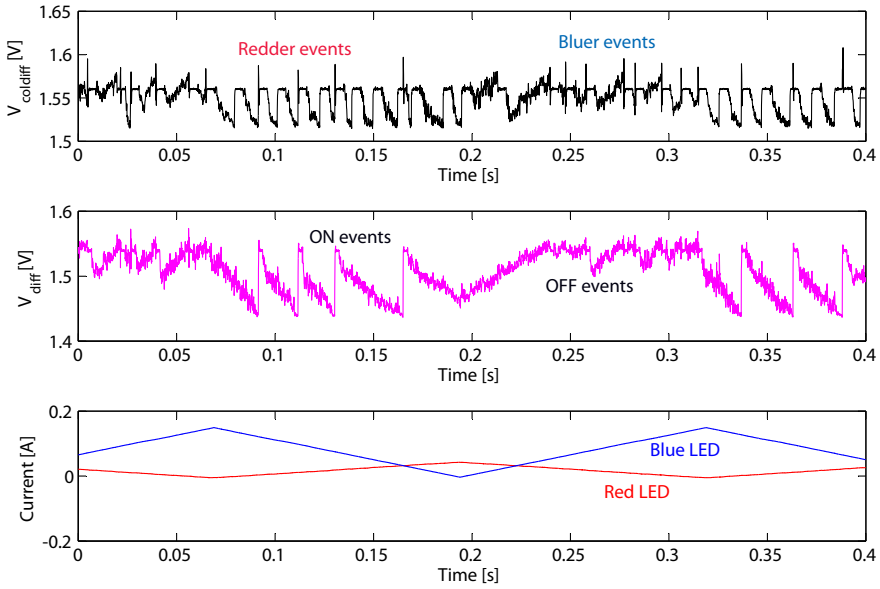


Figure 8.18: Test-pixel color sensitivity with ND2 filter

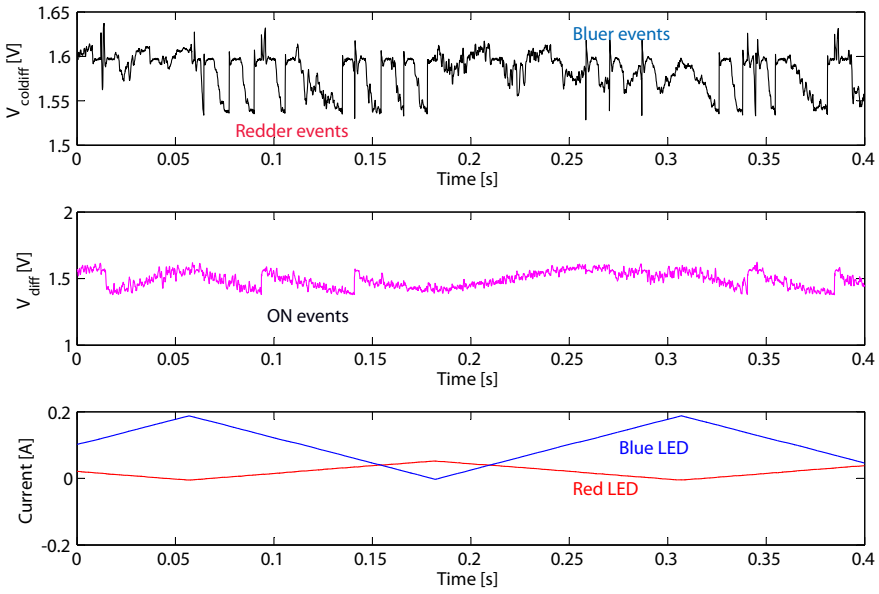


Figure 8.19: Test-pixel color sensitivity with ND3 filter

8.5.2 logCDVS Array

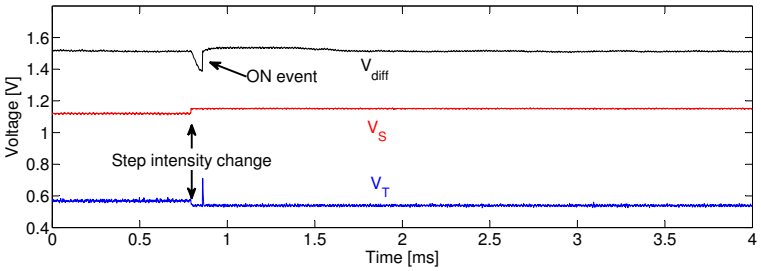
In the following we provide measurement results for the 32×32 array of logCDVS pixels. The array needs a considerable amount of light to avoid the coupling illustrated in Fig. 8.20. If the chip is not illuminated strong enough, the whole array is emitting events in an uncontrollable way.

Array color sensitivity and threshold mismatch

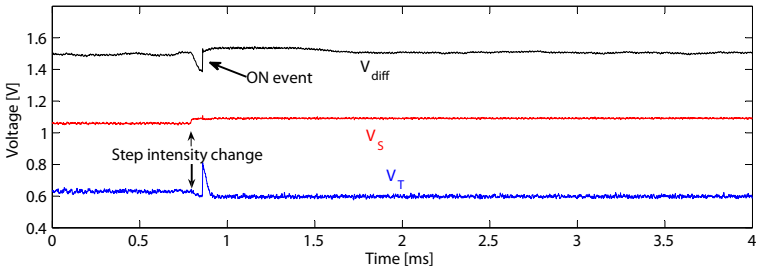
In the following experiments, the chip was stimulated with a color sweep from red to blue and vice versa similar to the test pixel experiment, but with sinusoidally varying currents. Fig. 8.22 shows a histogram of the number of events per pixel per cycle for a full color sweep and the rolling event histogram of each type (BLUER, REDDER, ON, OFF) during three color sweep cycles. Stimulating frequency is 1.5 Hz.

When recording spikes from the array rather than the test pixel, we need to set the threshold much farther away from the balance level to prevent a majority of pixels from spiking continuously. The sensitivity of the array to color change is therefore reduced significantly compared to the test-pixel. Even so there were still some pixels spiking continuously. These pixels are not counted for the mean number of events per pixel per cycle and the rolling event histograms.

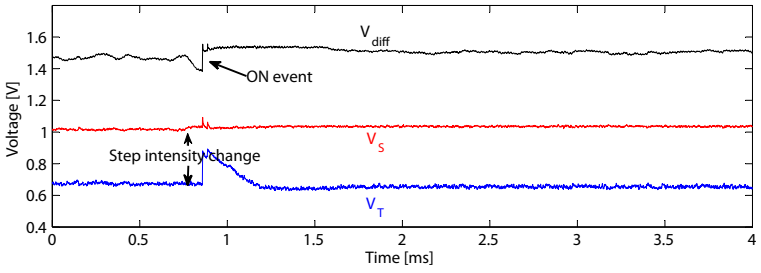
Fig. 8.23 shows the same color sweep experiment with a higher threshold setting, Fig. 8.24 with a lower threshold setting. Even with the low threshold setting, where 75 pixels were continuously emitting events and thus had to be suppressed



(a) No ND filter



(b) ND1 filter



(c) ND2 filter

Figure 8.20: Coupling from V_{diff} to the front-end.

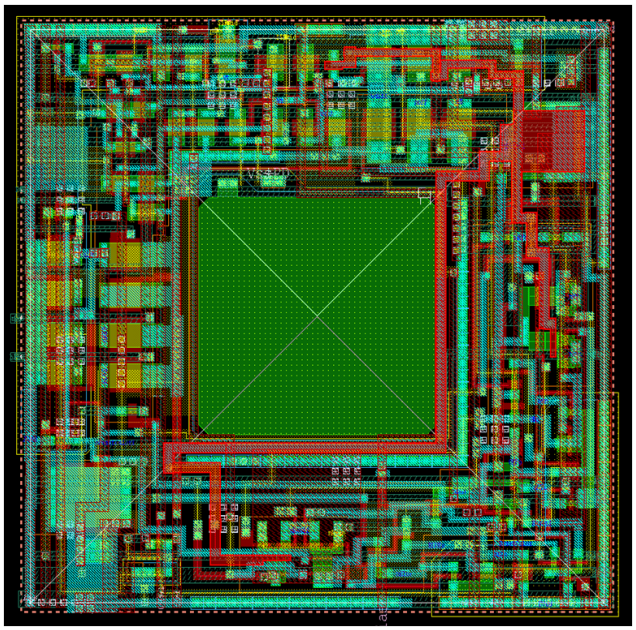


Figure 8.21: cDVSTest30 pixel layout with highlighted *nReseti* node.

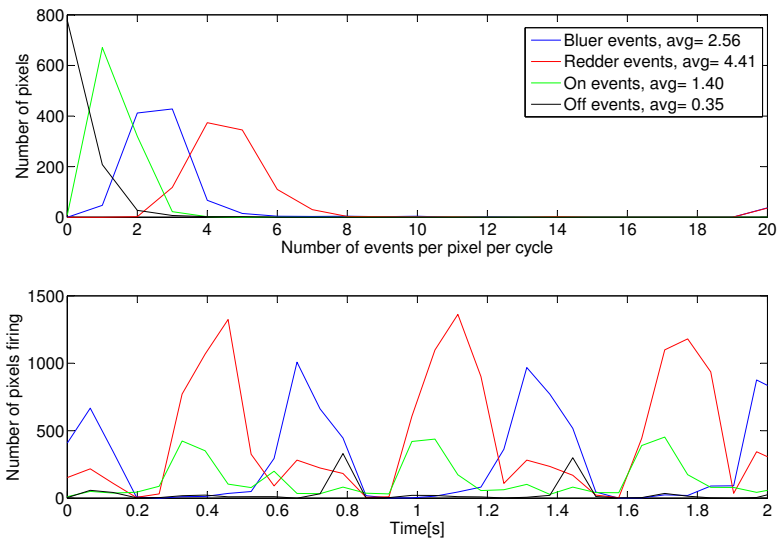


Figure 8.22: Response of the cDVS pixel to a stimulus with changing color from full red to full blue. The upper plot shows a histogram of the number of events per pixel per cycle of the input stimulus. The lower plot shows the number of events per 40ms bin over a duration of 2 seconds. Stimulus frequency is 1.5 Hz.

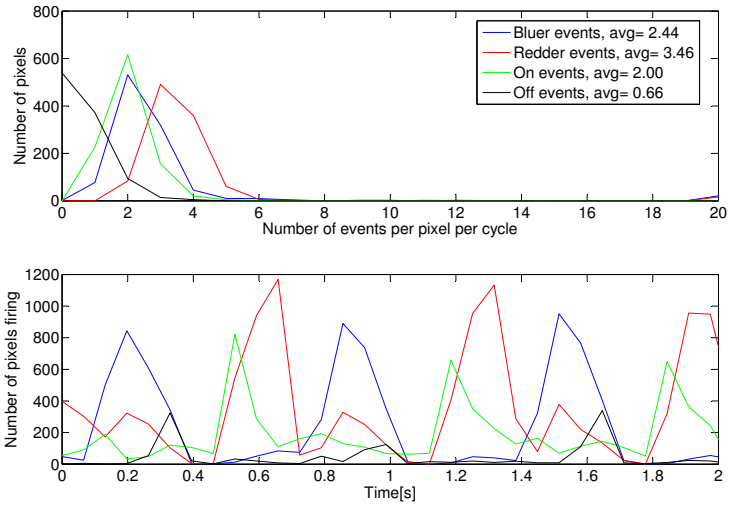


Figure 8.23: Response of the cDVS pixel to a stimulus with changing color from full red to full blue. High threshold setting.

in software, the pixels on average only produce 3 BLUER events and close to 6 REDDER events per cycle. The asymmetry is due to the strong leaking at high illumination, with an ND1 filter the response is symmetric four BLUER and four REDDER spikes per cycle. Color sensitivity is thus about 45nm wavelength change.

Sensitivity to intensity change

If the chip is stimulated with a sinusoidally varying intensity of a factor of three generated by a sinusoidal current in the red LED, the response of the color change pathway of a pixel is on average less than an event per cycle as illustrated in Fig. 8.25. Here the response of the intensity pathway is very asymmetric, but these thresholds could easily be tweaked to a more symmetric response.

Log intensity readout

The log intensity readout of cDVSTest30 is functional. Fig. 8.26 shows the chip observing a computer screen in front of a window, outside a bush and a building are visible. The chip observes the scene with a Computar CS-mount 2.6mm F1.6 lens. The illumination of the computer screen is around 1klux, outside in the sunlight it is several 10klux. Without calibration the computer screen is just barely visible, but image quality is very poor. With on- or off-chip calibration, image quality is improved.

In Fig. 8.27, the chip is observing a Kodak Gray Scale density step chart. The scene was illuminated with indirect sunlight, illumination was around 2.2klux. The

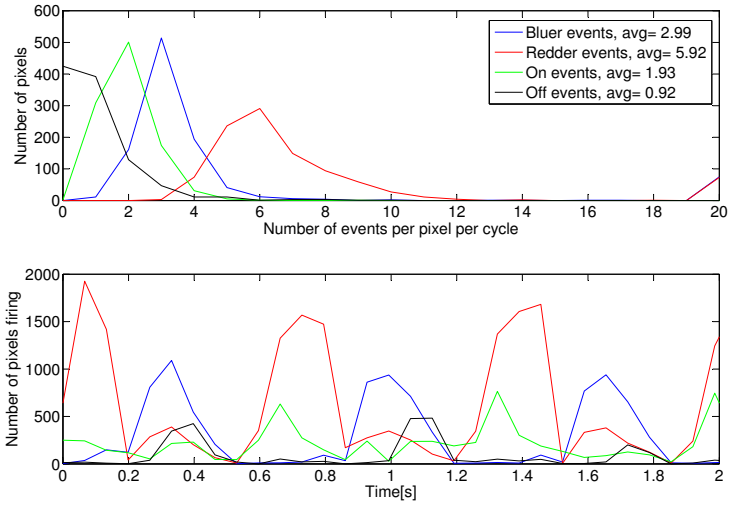


Figure 8.24: Response of the cDVS pixel to a stimulus with changing color from full red to full blue. Low threshold setting.

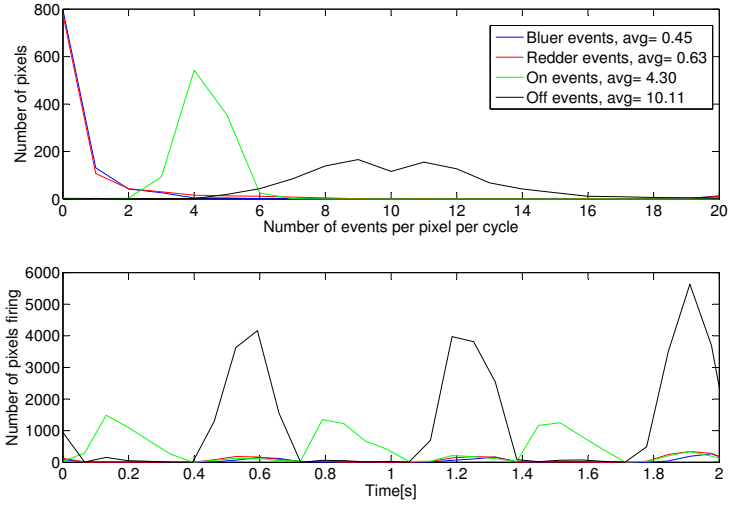
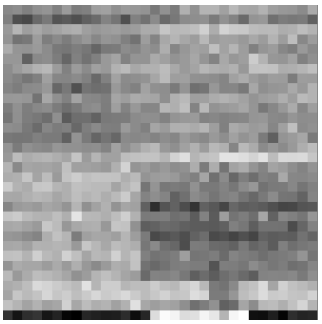
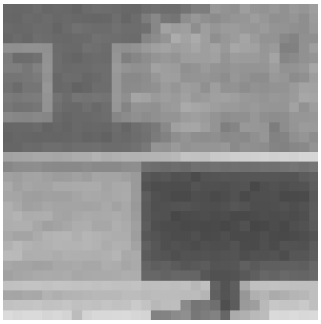


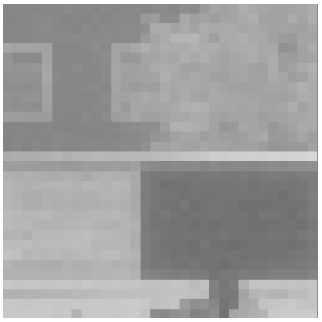
Figure 8.25: Response of the cDVS pixel to a stimulus with sinusoidally changing intensity and constant red color. The intensity change is a factor of three.



(a) No calibration



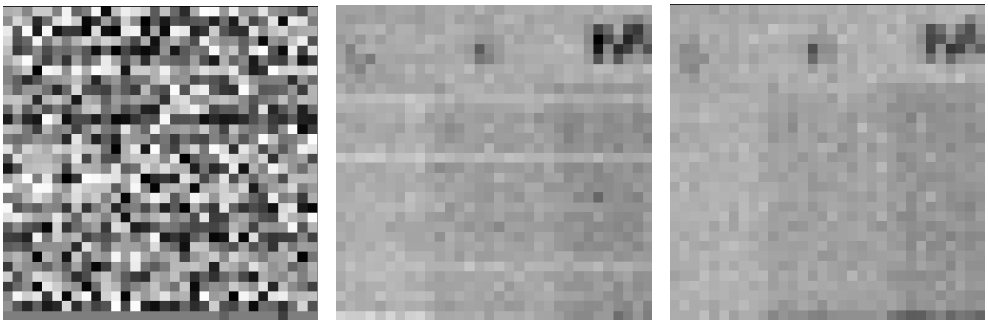
(b) On chip calibration



(c) Off chip calibration by subtraction of a reference frame

(d) Picture of observed scene.

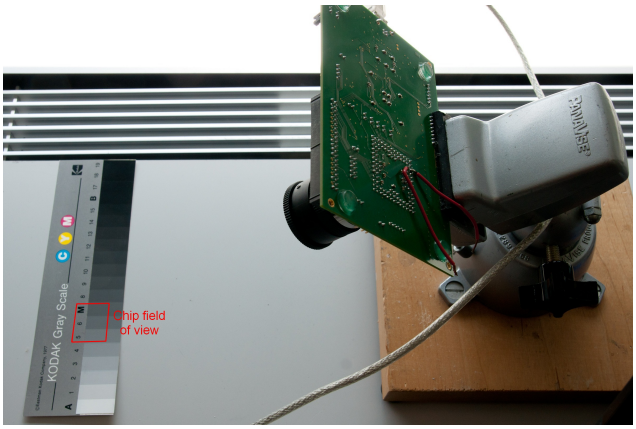
Figure 8.26: Log intensity pathway observing a computer screen in front of a window, outside a building with two windows is visible.



(a) No calibration

(b) On chip calibration

(c) Off chip calibration by subtraction of a reference frame

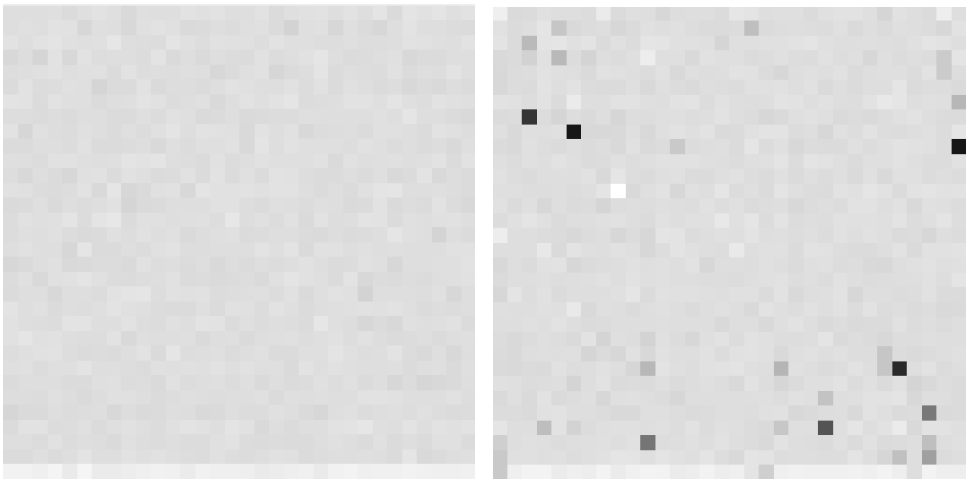


(d) Picture of chip observing density scale. The field of view of the chip is illustrated by a red polygon.

Figure 8.27: Log intensity pathway observing Kodak step chart.

chip was observing the scene with a Computar 8.5mm F1.3 lens. Without calibration, the output is unusable. With on-chip calibration, the 10% density steps are just barely discernable. With off-chip calibration (the reference frame was taken when the chip was observing the same scene without a lens), the 10% steps are visible.

When the array is not held in reset, event-generating pixels affect the log-intensity readout. This effect is illustrated in Fig. 8.28. The chip is observing a uniform scene, on-chip calibration is enabled. In Fig. 8.28(a) all the dynamic pathways are held in reset and in Fig. 8.28(b) the pixels are allowed to emit events. It is evident that the shielding from the event-generation pathway to the log-intensity readout is not sufficient. If the array is held in reset, the average difference between maximal and minimal value in a frame (averaged over 20 frames) is 44mV, standard deviation of the pixel values within a frame is 5.85mV. If the array is spiking, the average difference between maximal and minimal value in a frame is 327mV,



(a) Pixel array is held in reset. (b) Pixel array is spiking.

Figure 8.28: Events affecting the log intensity readout. On-chip FPN suppression is enabled.

standard deviation is 13.87mV.

To quantitatively characterize the log intensity readout and the FPN suppression, we stimulated the chip with constant light from the red LED and changed the light intensity with Kodak Wratten neutral density filters. The dynamic pathways are prevented from spiking by holding all the communication circuits in reset.

The output voltage of the chip is converted with a 12 bit AnalogDevices AD7933 analog to digital converter. The input signal range of the converter is 2.5V, which means the bit resolution is 0.61mV.

Figures 8.29 to 8.34 show measurement results of output voltage, fixed-pattern noise, temporal noise and contrast sensitivity for five measurement conditions:

1. On-chip calibration off, 5T buffer.
2. On-chip calibration off, source follower buffer.
3. On-chip calibration on, 5T buffer.
4. On-chip calibration on, source follower buffer.
5. Off-chip calibration by subtracting a reference frame, 5T buffer.

For all the measurements, the outermost rows and columns are neglected, and so the measurements present results for a 30×30 pixel array. The reference frame for off-chip calibration was recorded with ambient light at 750 lux, $1.7W/m^2$.

Fig. 8.29 shows the mean pixel output voltage of the chip for four decades of intensity. It is clearly seen that the difference amplifier for the on-chip calibration provides gain and therefore higher output swing, but it also decreases linearity.

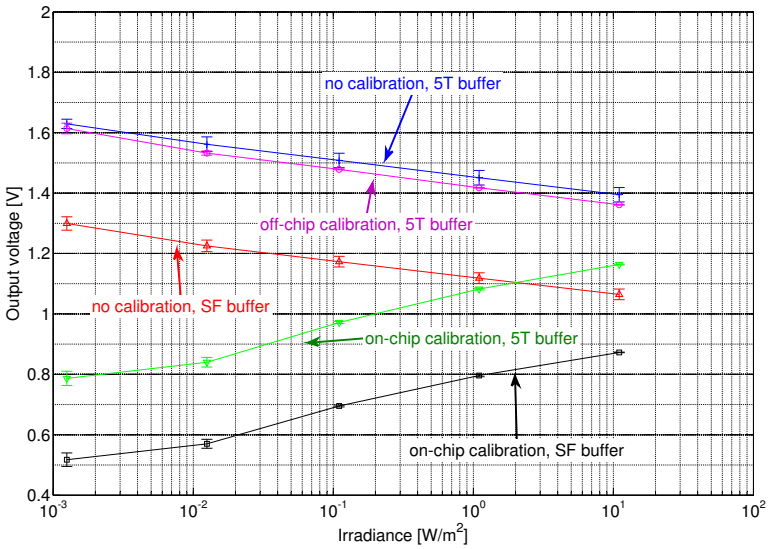


Figure 8.29: Mean logarithmic output voltage over 4 decades of light intensity

Measurement condition	Gain [mV per decade]
On-chip calibration off, 5T buffer	55.7
On-chip calibration off, source follower buffer	53.5
On-chip calibration on, 5T buffer	107.9
On-chip calibration on, source follower buffer	100.9
Off-chip calibration, 5T buffer	56

Table 8.3: Gain per decade of the logarithmic intensity readout

Especially at the lowest intensity the gain is reduced, which is probably because the pixels do not have enough time to slew back to the photocurrent output level after switching off the calibration current in the last frame. This problem is know as lag, because the last frame influences the current frame. Decreasing the frame rate might solve this problem.

The mean gain per decade of light intensity is shown in table 8.3. Using a 5T buffer decreases gain variation between the rows as illustrated in Fig. 8.30. Even for high illumination, the standard deviation of the pixel gain is several percent and increases a lot for low photocurrents. A two-point calibration that is able to correct both offset and gain variation would thus be beneficial.

The 5T buffer provides between five to six percent higher gain than the source follower buffer, at the expense of vastly increased FPN, which is illustrated in Fig. 8.31. The on-chip calibration is able to suppress the FPN to below 10% con-

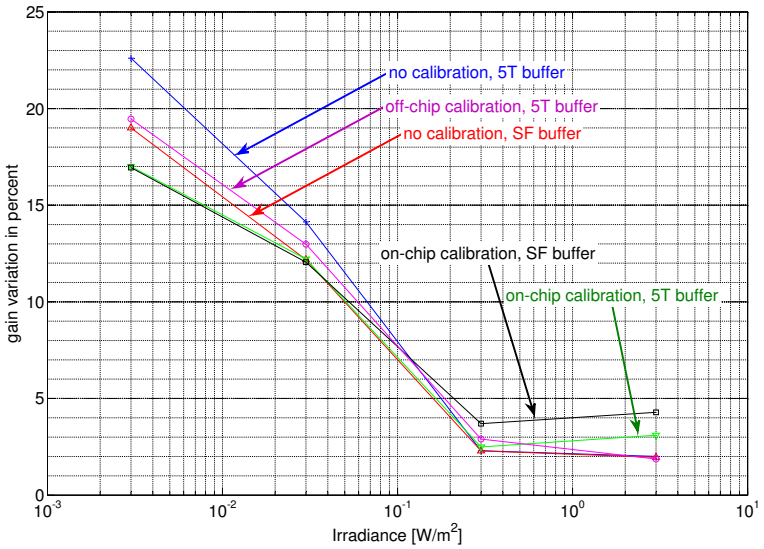


Figure 8.30: Gain variaton

trast for chip illumination above $0.1W/m^2$. For lower intensities, on-chip FPN suppression is much less efficient than off-chip FPN supression.

Fig. 8.32 illustrates the FPN within a row. On-chip calibration is able to suppress FPN below 3% contrast at high intensities. Most of the chip FPN is caused by the row buffers.

Fig. 8.33 shows the temporal noise. The temporal noise when on-chip calibration is switched on is roughly a factor 2.5 higher than with calibration switched off, which is more than the amplification by the difference amplifier. Signal-to-noise ratio (without considering FPN) is thus better when on-chip calibration is switched off.

Total contrast sensitivity is defined by the minimal visible contrast in a scene. Both fixed pattern noise and temporal noise limit the minimal contrast that can be discerned reliably. Fig. 8.34 shows contrast sensitivity for the cDVSTest30 logarithmic output. In our case, FPN clearly limits contrast sensitivity, thus Fig. 8.34 strongly resembles Fig. 8.31. We see that using a 5T buffer brings only a slight advantage over a source follower buffer.

8.6 Discussion

Limited supply voltage and the necessity to keep both junctions of the BDJ reverse biased limit the voltage headroom for the intensity readout. It is necessary to keep both feedback transistors M_{N1} and M_{P1}, the cascode transistor M_{RCas} as well as the readout transistor $M_{\log}$ in saturation for correct functioning.

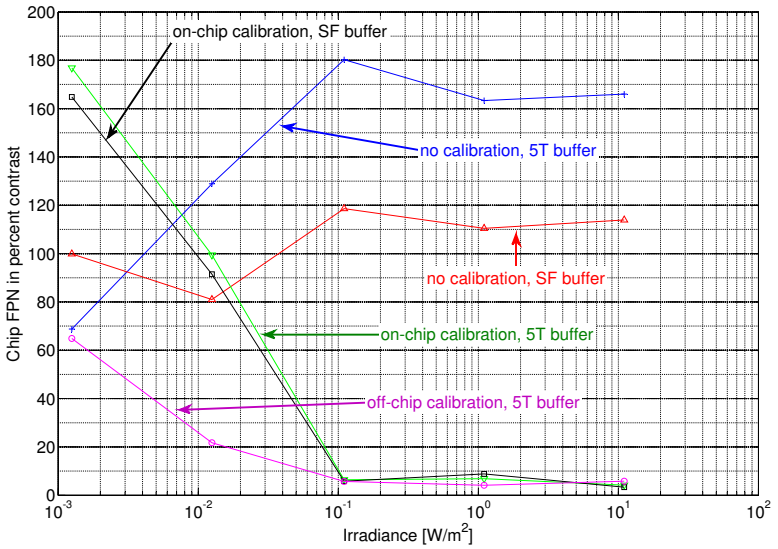


Figure 8.31: FPN across the array over 4 decades of light intensity. The FPN is expressed in percent contrast, a value of 10% means that the FPN has an amplitude equal to a stimulus with 10% contrast. In this example of 10% FPN, a pixel that is illuminated 10% stronger than its neighbors can thus not reliably be distishuished because of the FPN.

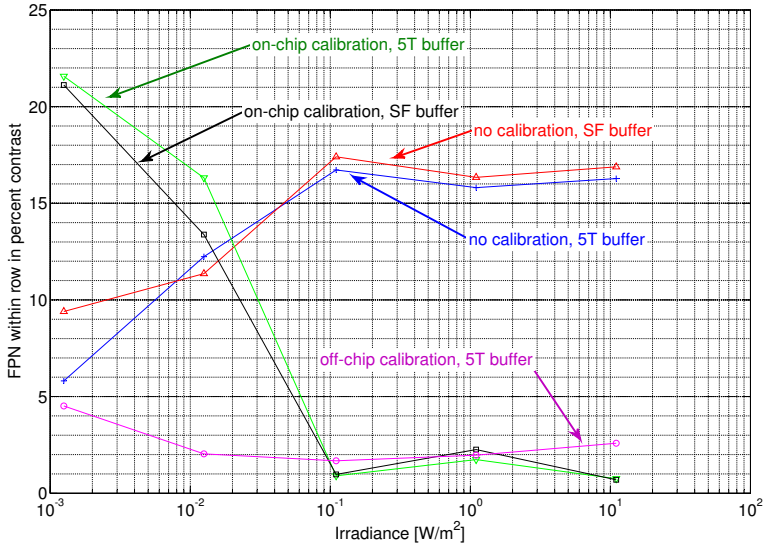


Figure 8.32: FPN within one row (which shares common row buffer readout circuitry) expressed in percent contrast over 4 decades of light intensity.

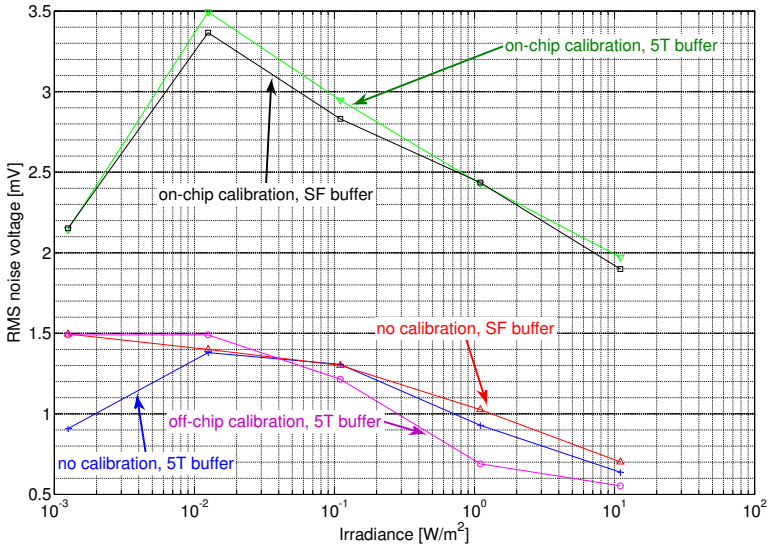


Figure 8.33: Output noise voltage over 4 decades of light intensity

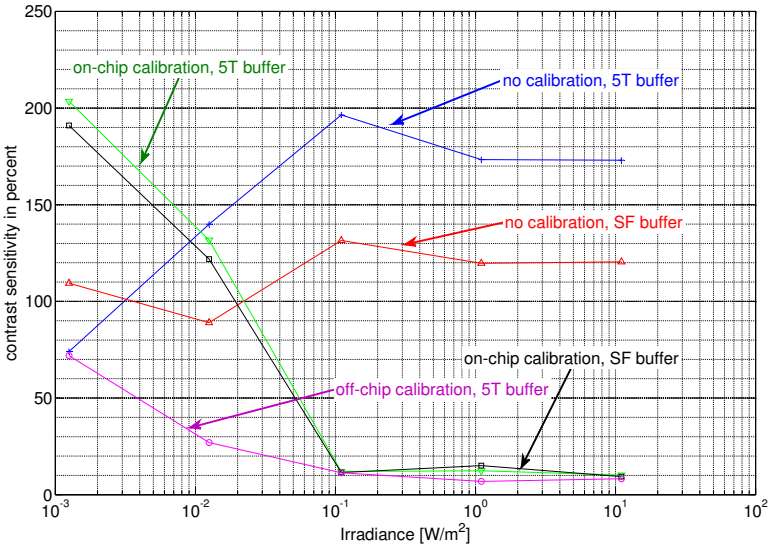


Figure 8.34: Contrast sensitivity expressed in percentage contrast over 4 decades of light intensity

This headroom limitation is problematic in the logCDVS pixel because it limits the range of possible photoreceptor bias currents. If the photoreceptor bias current is too high, the feedback transistor M_{N1} goes out of saturation. For this limited power supply of 1.8V, using a double-photoreceptor BDJ front-end does not leave enough headroom for any kind of voltage-mode readout. A solution would be to run the front-end at 3.3V, which would increase layout area and power consumption.

8.6.1 Color change detection

With cDVSTest30, we demonstrate the first multi-pixel event-based color vision sensor. But this chip does not provide a performance which allows to scale up the array and use it for applications. Color sensitivity is low and the chip needs high illumination to work properly because of coupling from one event-generation pathway through the photoreceptors to the second event-generation pathway.

To decrease the coupling from the event-generation pathways to the photoreceptors, more careful layout is needed to reduce parasitic capacitances. Also the source follower transistors M_{N4}, M_{P4} (Fig. 8.1) and M_{N4i} (Fig. 6.5) should be made narrower to decrease the gate-source overlap capacitance.

Using a two-stage amplifier like cDVSTest10 (Chapter 6) with reduced and more carefully chosen gain and improved reset circuit will improve the color sensitivity.

8.6.2 Log-intensity read-out

The log-intensity read-out is functional, but contrast sensitivity is low for low to intermediate illumination because of high FPN. On-chip FPN suppression is not very effective at low intensities due to gain mismatch, the FPN suppression circuits only suppress offset mismatch. The FPN suppression is more efficient at higher intensities because the photocurrents are closer to the calibration current and thus gain mismatch has less influence. For low intensities the photocurrents and the calibration current are several decades apart. Possibly the FPN suppression could be improved for lower intensities by dynamically adjusting the calibration current. However, off-chip offset calibration by subtraction of a reference frame recorded at moderate intensity yields acceptable performance over several decades of illumination and is thus to be favored.

Signal quality is highly degraded if the dynamic pathways are emitting events. To decrease the coupling from event generation pathways to the intensity readout, the source follower transistors from the photoreceptors to the dynamic pathways should be made narrower to reduce capacitive coupling from the differentiators to the photoreceptors.

The buffer transistor M_{RO} is quite small ($W/L = 0.46\mu m/0.4\mu m$, roughly $0.4fF$ gate capacitance) in the current design. It would be beneficial to increase the size of this transistor to increase the parasitic capacitance of the log-intensity node $V_{\log}$, which will make this node stiffer at the expense of increasing lag for very low intensities.

Using a 5T row buffer instead of a source follower buffer increases FPN significantly if no correction is applied. The on-chip FPN correction is able to decrease the FPN to a level which is very similar to the source follower buffer. Contrast sensitivity is slightly better, because the 5T buffer has slightly more gain at the expense of increased complexity. Whether a 5T buffer or a source follower buffer is more suitable depends if the additional complexity (one more metal line to the pixel, three more transistors in the periphery) can be tolerated. In the case of the logCDVS pixel, which is already quite complex, an additional metal line can be fit without problems and also on the periphery enough space to fit the additional transistors is available due to the relatively high pixel pitch of $29\mu m$.

The transistors M_{invN} and M_{invP}, which form a NAND operation of the *ColSel* and *nRefEnable* signals, are completely redundant and should be removed. This NAND operation is already done in the periphery at the bottom of each column.

Chapter 9

Conclusion and Retrospective analysis

In this chapter I summarize the work described in this thesis, relate to the state-of-the-art and propose modifications and concepts for further research directions. My work improves previous work on event-based vision sensors in three different areas:

- Color change detection using buried double junctions. We present the first event-based asynchronous color vision sensor.
- Combination of dynamic and sustained pathways in a single sensor. We present a novel and compact way of combining asynchronous change detection circuits with synchronous frame-based intensity read-out.
- Word-serial Address Event Representation communication circuits. We present a modified word-serial AER protocol that reduces the number of transistors per pixel compared to previous work.

9.1 Color change detection

Chapter 4 presents novel pixel circuits which are able to detect wavelength changes. The most promising of this pixel architectures was selected, evolved and a small array of 32×32 pixels combining color change detection with log intensity change detection was fabricated. This thesis thus presents the first event-based asynchronous multi-pixel color vision sensor. Previous work only showed a single pixel design [83].

The color change detection pixel array is functional and shows a proof of concept, but at this stage the pixel design is not usable for applications. Color sensitivity of a single test-pixel is encouraging, but tuning of the 32×32 array did not allow for usable sensitivity. The gain of the single stage amplifier of cDVSTest30 is not enough to overcome the mismatch in the comparators. Increasing the gain in a

single stage amplifier is not practical due to size constraints for the capacitors. Using a two-stage amplifier as initially done leads to other kinds of problems as demonstrated in Chapter 6, but might give usable results if the gain of each stage is more carefully chosen and the reset circuit improved. However, a higher sensitivity than detecting wavelength changes of around 15nm (12 events for a red to blue sweep) can probably not be expected.

If higher color sensitivity is desired, the UMC 180nm process used to build cDVSTest30 is not suitable. Other CMOS process technologies may offer better sensitivity, but unfortunately we are not able to predict the color sensitivity of a BDJ based on the assumption of step junctions given doping concentrations and junction depths.

Building an event-based color silicon retina with high color sensitivity should probably be intended with a CMOS image sensor process which provides access to color filters, at the expense of spatial color aliasing, light loss and higher prototyping costs. But different pixel architectures are needed for event-based color sensitive pixels built with color filters. Before investigating transistor level circuit design and fabricating prototype chips at higher cost, possible pixel functionality should be modeled and the usefulness of the output evaluated.

9.2 Combination of dynamic and sustained pathway

In Chapter 8, a novel way of combining dynamic event based change-detection pixels with measuring intensity at the same spatial location is presented. This circuit uses the same photocurrent for the dynamic and the sustained pathway. This can reduce pixel size compared to the previous design by Posch et al. [67], or, as done in this thesis, allows to include more functionality in the same pixel size.

Our design does not reach the performance of Posch's design. Signal swing is limited and fixed-pattern noise is acceptable only for high intensities even with FPN suppression. The color change detection pathway limits the available headroom for the intensity readout, because both junctions of the BDJ need to be reverse biased. A combination of the intensity readout and log intensity change detection with a normal well-substrate photodiode gives more headroom for the intensity readout. More headroom allows using an additional diode-connected transistor M_d as illustrated in Fig. 9.1 to increase signal swing and thus signal-to-noise ratio. Using this double-diode current-to-voltage conversion will probably introduce more gain variation. However, the relative FPN contribution of the row buffers would decrease due to increased signal swing. In such a design a two point correction as proposed by Choubey et al. [138] would be beneficial to achieve acceptable contrast sensitivity. This two-point correction can not easily be done on-chip.

Using an integrating scheme will most likely yield better performance and eliminate lag (memory of past samples). That could be linear integration as in a standard APS pixel if intra-scene dynamic range of less than 60dB is acceptable. If higher intra-scene dynamic range is desired, a stepped reset-gate voltage can be

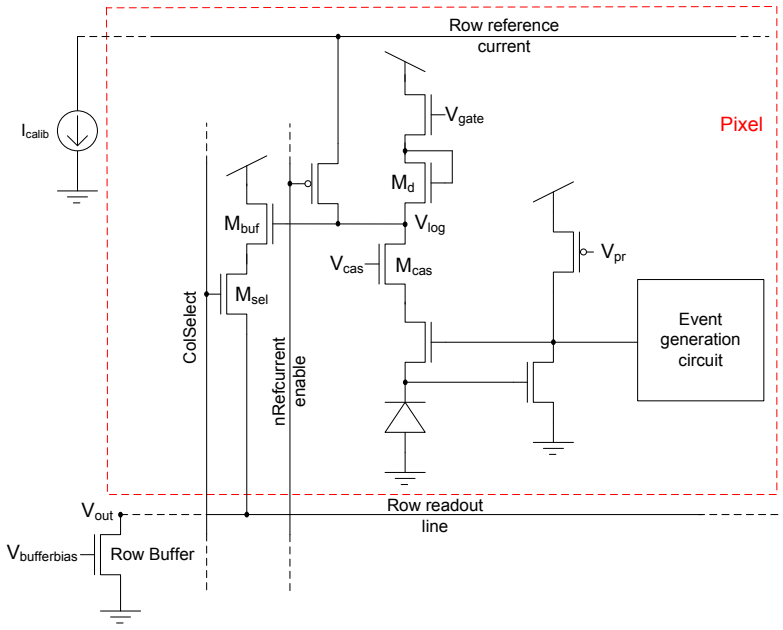


Figure 9.1: DVS pixel with log intensity readout with increased signal swing

used without adding complexity to the pixel [45].

Fig. 9.2 illustrates the combination of a DVS pixel with an APS circuit. The APS circuit adds four transistors ($M_{\text{buf}}, M_{\text{cas}}, M_{\text{sel}}$ and M_{reset}) to the DVS pixel which allows a very compact implementation. An advantage compared to conventional APS pixels is that the photocurrent does not have to charge the relatively large parasitic diode capacitance, but only the smaller gate capacitance of the readout buffer transistor M_{buf}. By choosing the size of M_{buf}, the sensitivity of the pixel can be adjusted.

In this proposed implementation, the reset switch M_{reset} is addressed in a column parallel manner to allow compatibility with most of the peripheral circuits of cDVSTest30. By switching the column reset from the power supply to some intermediate voltage (instead of ground as in a normal APS), transistor M_{reset} limits the maximal voltage swing at the integration node to avoid interfering with the event generation circuit. An additional benefit of this is a softer shoulder for clipped highlights for pixels that saturate. The signals in the APS-DVS are illustrated in Fig. 9.3.

The DVS test pixels in cDVSTest30 have a pixel pitch of $14.5\mu m$ and a fill-factor of 15%. I am confident that the four additional transistors needed for the APS-DVS can be added without significantly increasing the pixel size. An APS-DVS pixel with a pixel pitch of $17\mu m$ and a fill-factor of 11-12% should be feasible, which would be less than one third of the area of an ATIS pixel [67], which is built in the same process technology and provides similar functionality of combining asynchronous

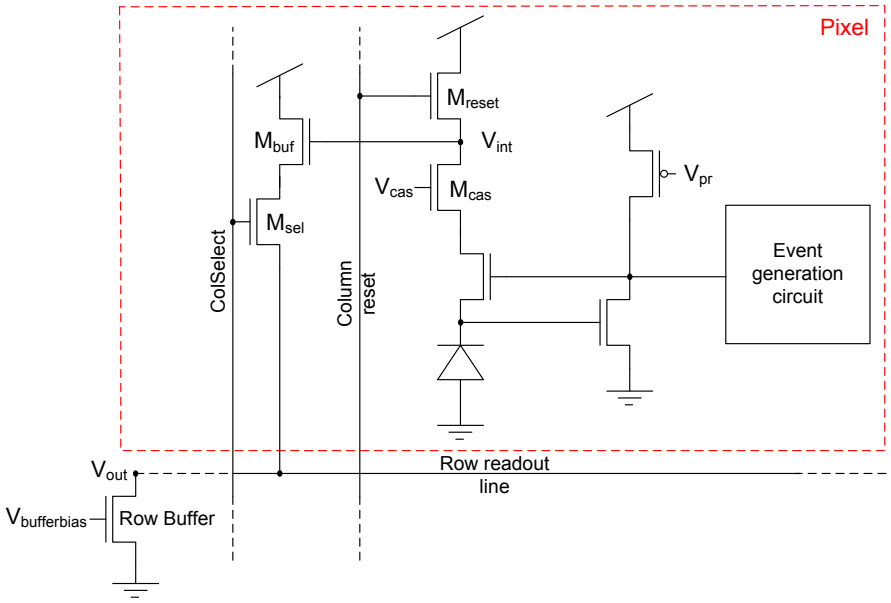


Figure 9.2: DVS pixel combined with APS circuit

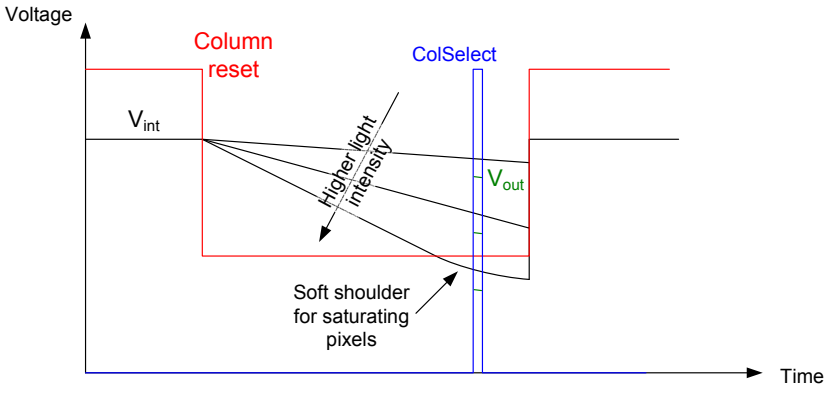


Figure 9.3: Signals in the APS-DVS

change detection and intensity information readout.

9.3 AER communication circuits

The realization of the new AER communication circuits presented in chapter 7 is an important step to higher resolution event-based vision sensors. The circuits have been inspired by the work presented by Boahen [31]. Our design needs less transistors in the pixel at the expense of increased complexity in the periphery and the introduction of a column acknowledge line.

Compared to the AER circuits used in the DVS128, the new circuits improve the communication bandwidth by more than an order of magnitude. Additionally the new circuits have the advantage that a malfunctioning pixel can not completely block the communication. In the DVS128, the pixel with the lowest threshold limits the contrast sensitivity, because it has to be taken into account when setting the event thresholds. In the new AER circuits, outlier pixels with very low thresholds can not constantly block the bus, instead they will emit events at a rate set by the refractory bias.

The communication bandwidth achieved with the new AER circuits compares favorably to the results published by Boahen [33], however Boahen's circuits are built in $0.25\mu m$ technology, increased speed of our circuits is thus to be expected just from technology scaling. Posch et al. [67] do not state the communication bandwidth of the ATIS, which would be interesting to compare because it uses very similar dynamic pixels and it is built in the same technology.

9.4 Closing words

Even though this thesis does not present a device that is usable in real-world applications, it provides important steps towards more powerful event-based vision sensors. The APS-DVS pixel combined with word-serial AER communication circuits will result in a high-performance vision sensor, combining sustained and dynamic pathways in a compact pixel. This allows higher resolution in the same die size or a much smaller and therefore cheaper die compared to the ATIS [67].

The outlook for event-based color vision sensors is not as bright though. Our novel BDJ pixel circuits did not yield the expected performance. To build a high quality event-based color vision sensor, new pixel architectures are needed. To benefit from an event-based readout, these new pixel architectures should emit informative events and reduce redundancy, as opposed to the pixel presented by Olsson et Häfliger [83], which provides little benefit compared to standard image sensors.

Appendix A

cDVSTest PCB

To test the cDVSTest30 chip and interface it to a computer, we developed a PCB platform that is easy to use, compact and portable. The cDVSTest PCB is based on the USBAERmini2 AER interface that I developed during my master thesis [30, 141].

A.1 Overview

Like the USBAERmini2, the new platform uses a Cypress FX2LP USB2 Highspeed transceiver for the connection to the PC. Due to the added complexity of the word serial AER protocol and the intensity readout, the Xilinx Coolrunner 2 CPLD used in the USBAERmini2 is not sufficient to interface to the cDVSTest30 chip, thus the Coolrunner 2 CPLD has been replaced by a Lattice MachXO CPLD. We chose the MachXO because it only needs 3.3V power supply, has an internal phase-locked loop (PLL) circuit to control the system clock frequency, internal flash memory to store the configuration and is available in a convenient package.

Additionally there is an AnalogDevices AD7933 analog to digital converter to convert the log intensity values of the cDVSTest30 chip. The PCB also hosts space for two AnalogDevices AD5391 digital-to-analog converters (DAC) to provide bias voltages in case the on-chip bias generator does not work properly. Because the bias generators work as desired, it was not necessary to mount these DACs. The cDVSTest PCB has two dual 3.3/1.8V regulators. One of the regulators is used to power the ADC (3.3V) and the analog parts of the chips (1.8V). The second regulator is used to power all the digital components of the PCB (3.3V) as well as the digital parts of the chip (3.3V for I/O and 1.8V for the core). Both regulators are powered through the USB bus. Fig. A.1 shows schematically how the components are connected, Fig. A.2 shows a picture of the cDVSTest PCB.

The PLL included in the MachXO is used to increase the clock frequency of the AER handshake interface. The first-in first-out (FIFO) buffer interface between CPLD and FX2 runs at 30MHz, but the rest of the internal circuitry of the CPLD including the handshaking runs at 90 MHz which increases the peak event rate to

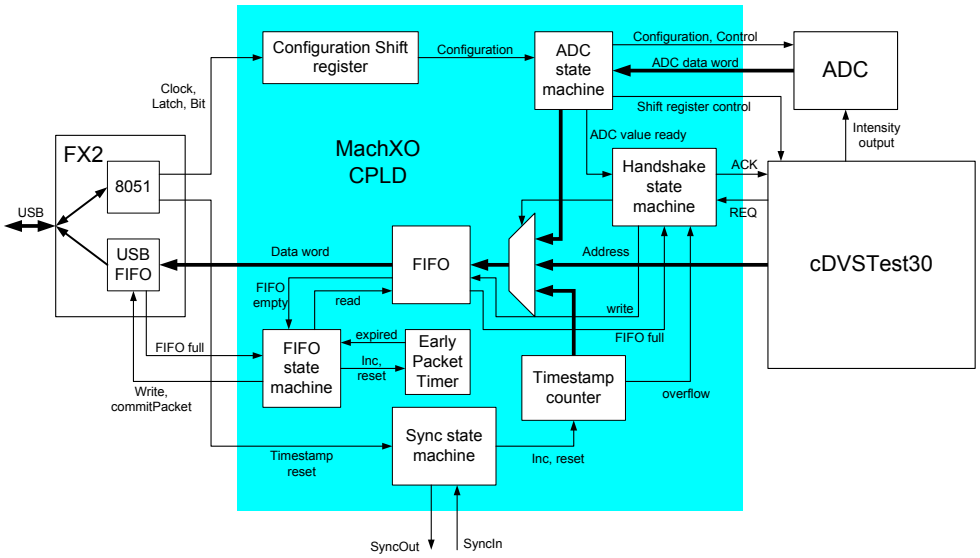


Figure A.1: cDVSTest PCB schematic

about 11 Meps (Mega-events per second), compared to 6 Meps of the USBAERmini2. The MachXO also includes block RAM which is used as an AER event FIFO to increase the time during which the peak event rate can be sustained.

A.2 CPLD firmware

The firmware of the CPLD is responsible for the following tasks:

- It handshakes with the chip, timestamps the events and writes time-stamp and address to the FX2 FIFO buffer.
- It controls the ADC. This includes writing configuration bits, starting the acquisition and storing the ADC value in a register.
- It controls the shift registers and reference current enabling of the cDVSTest30 chip. The shift registers have to be loaded with appropriate bits and clocked to select the correct pixel.

The components of the firmware are described in the following subsections, Fig. A.1 illustrates how they are connected. The components are mostly based on the USBAERmini2 CPLD firmware which is described in detail in my master thesis [141].

A.2.1 CPLD internal configuration shift register

Because this firmware includes several parameters which should be under host computer control, we implemented a configuration shift register. This shift register is

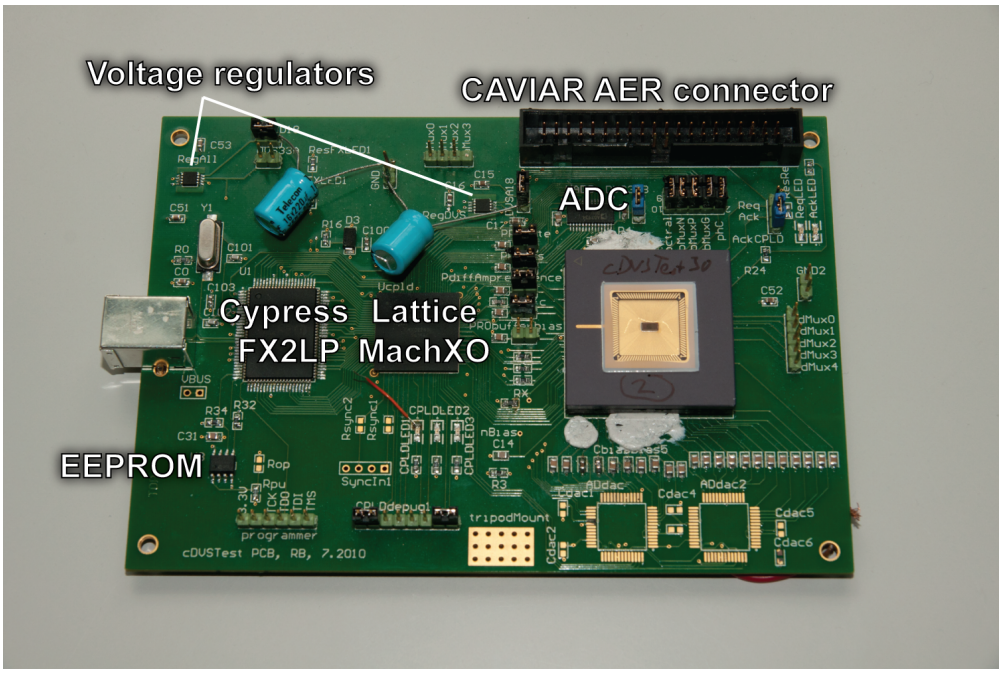


Figure A.2: cDVSTest PCB

loaded from the 8051 micro-processor core inside the Cypress FX2, which receives the data from the host computer. Table A.1 lists the parameters which are stored in the shift register.

The shift register is controlled with three pins: *Clock*, *BitIn* and *Latch*. During normal operation, *Latch* is held low to store the current configuration. To load a new configuration, the new bits are clocked in using *Clock* and *BitIn*. When all the bits are loaded, *Latch* is asserted which copies the bits from the shift registers to the storage elements.

A.2.2 Hand-shaking state machine

The hand-shaking state machine implements the four-phase handshaking with the sender chip and writes the corresponding addresses and timestamps to the CPLD internal FIFO. It has to distinguish between column and row addresses, because only for row addresses it is necessary to write a time-stamp to the FIFO.

This state machine, illustrated in Fig. A.3, additionally writes ADC values, time-stamp counter wrap events (see section A.4) and reset events to the FIFO. Every data word written to the FIFO consists of two type bits and 14 data bits. The different types are listed in table A.2.

Handshaking with the sender needs three or four state transitions, depending whether it is a column or row address. This could be lowered to two or three state transitions by including a pipelining register for storing the address before writing

Parameter name	size (bits)	description
Sel5T	1	Whether 5T-buffer or source-follower is used for log-intensity read-out.
UseCalibration	1	Using on-chip mismatch calibration
ADCconfig	12	ADC configuration bits
TrackTime	16	Settling time for log read-out (phase 1 of difference amplifier sequence, see section 8.3.2)
RefOnTime	16	Settling time for reference current read-out (phase 2 of difference amplifier sequence) Only used in on-chip calibration mode.
RefOffTime	16	Settling time of difference amplifier output (phase 3 of difference amplifier sequence) Only used in on-chip calibration mode.
IdleTime	16	Idle time between ADC conversion and selection of next pixel
ScanY	5	Row for single pixel conversion mode
ScanX	5	Column for single pixel conversion mode
ScanEnable	1	Full array or single pixel ADC conversion

Table A.1: CPLD shift register parameters

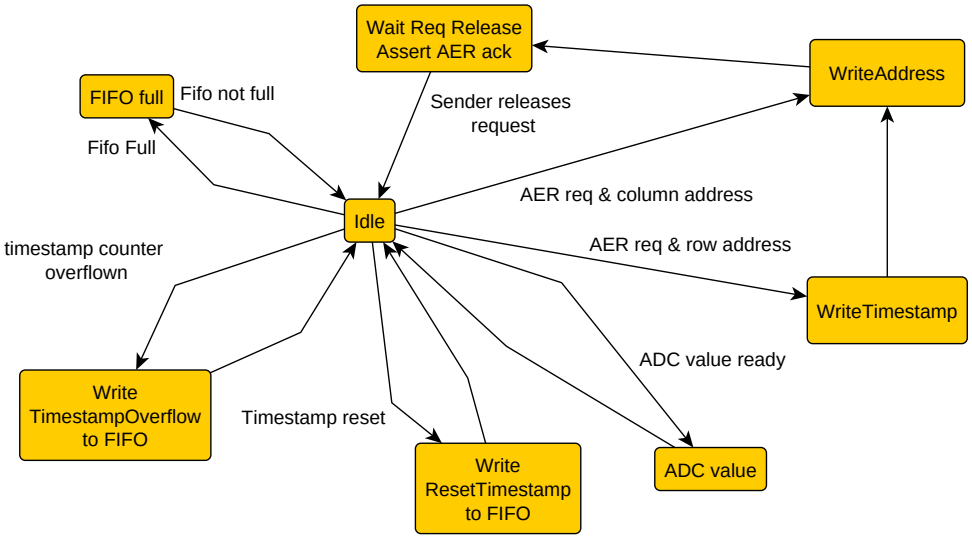


Figure A.3: Hand-shaking state machine

Type value	Description
00	Address or ADC value. The next bit distinguishes between ADC-value and event address, '1' for ADC, '0' for address.
01	Time-stamp
10	Time-stamp wrap
11	Time-stamp reset

Table A.2: Data types for cDVSTest

to the FIFO and checking whether the sender has already released *Request* in the WriteAddress state. Additional to the clock cycles needed for the state transitions, the asynchronous *Request* signal has to be synchronized to avoid meta-stability, which is done by two flip-flops in series. This will also take about two clock cycles per edge, so the full handshaking will take 7 or 8 clock cycles, which amounts to a peak event rate of about 12 Mega-events per second. To decrease the number of clock cycles needed for synchronizing the *Request* signal, the synchronization flip-flops could be clocked at a higher frequency.

A.2.3 ADC state machine

The ADC state machine handles all the communication with the ADC, controls the pixel-selection shift registers in the cDVSTest30 chip and also the on-chip mismatch correction difference amplifier.

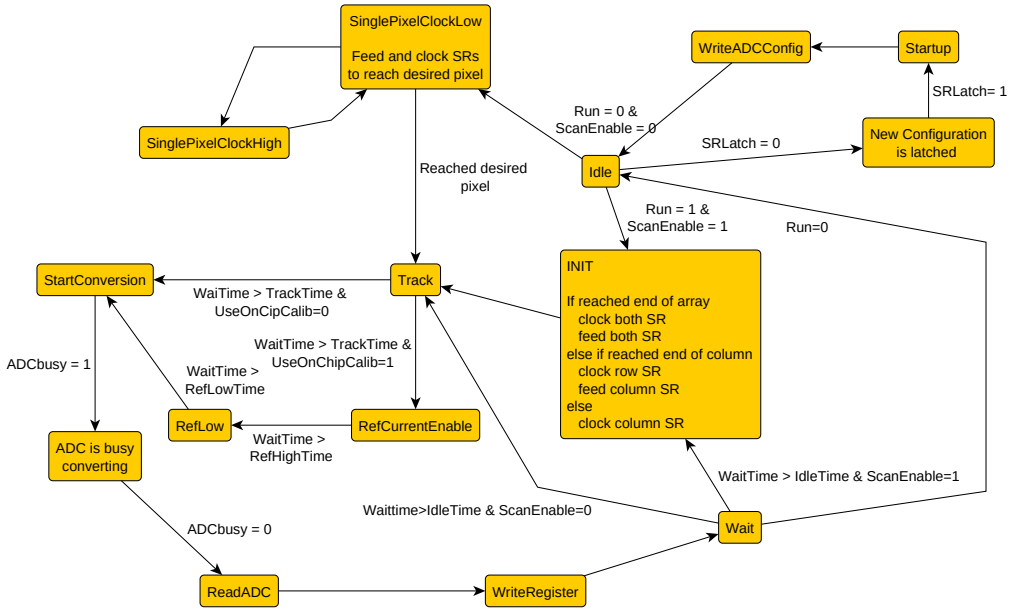


Figure A.4: ADC state machine

The state machine has several parameters which are stored in the shift register (table A.1). The on-chip mismatch correction can be enabled or disabled. The ADC state machine can be configured to either scan sequentially through all the pixels, recording a full picture or to continually monitor a single pixel.

The state machine has to remember which pixel it is currently converting, set the "Start-Bit" in the ADC data word for the first pixel in the frame for correct display on the host computer, and feed the pixel selection shift register with appropriate bits.

Contrary to most standard image sensors, the cDVSTest30 is scanned vertically instead of horizontally. At the end of each column, the row selection shift register has to be fed with a '1' and the column selection shift register clocked to select the next column. At the end of the array, both shift registers have to be fed with a '1'.

A.2.4 Synchronizer state machine

The synchronizer state machine controls the time-stamp counter. It offers the possibility to synchronize two boards by means of an electrical connection, in this mode one of the boards acts as master and controls the time-stamp counter of the second (slave) board.

In master mode, the counter is increased once per microsecond, so every 90 clock cycles.

In slave mode, the device increases the counter for every down-going pulse on the SyncIn input. If the SyncIn input is inactive for more than two clock cycles,

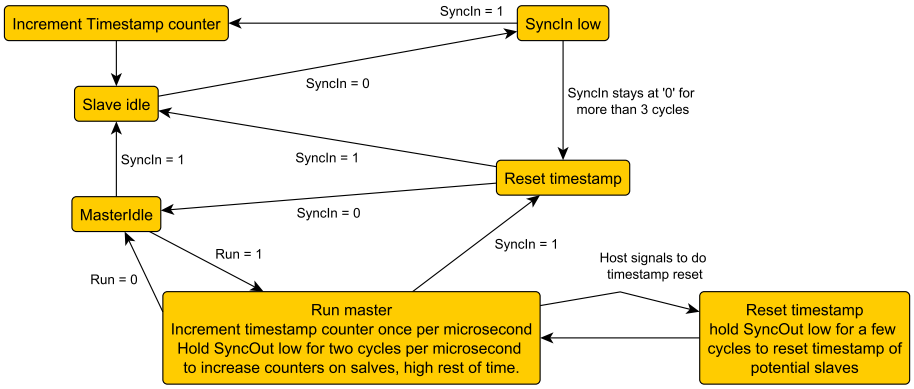


Figure A.5: synchronizer state machine

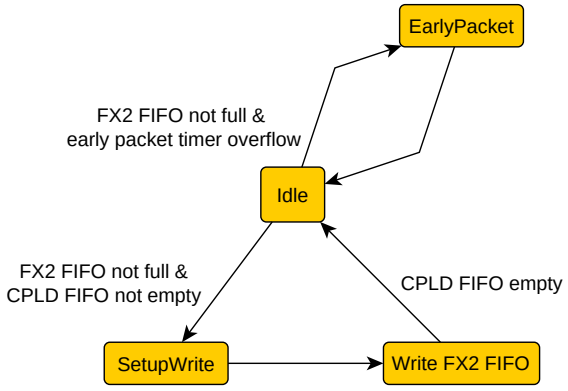


Figure A.6: FIFO state machine

the device resets the time-stamp and changes back to time-stamp master mode.

More details about the synchronization feature can be found in the publications about the USBAERmini2 [30, 141].

A.2.5 FIFO state machine

This state machine interfaces the CPLD to the USB FIFOs of the FX2LP. It reads the data from the CPLD-internal FIFO and writes them to the FX2. It acts as interface between the 90MHz clock domain and the 30MHz clock domain of the FX2LP FIFO interface.

Early packet timer

To ensure a minimal packet rate over USB, which is necessary for real time robotics, the early packet timer ensures that at least every few milliseconds a packet is sent

to the host computer, even if the packet is not full. Every time a full packet is sent, the timer is reset to zero.

A.3 Cypress FX2LP firmware

The Cypress FX2LP is an USB2.0 transceiver with an enhanced 8051 micro-controller and a flexible interface to its 4 kilobyte of first-in first-out (FIFO) buffers, which are committed automatically from or to the USB domain by the Cypress serial interface engine (SIE). We use the FX2LP in its slave FIFO mode, which means that the device handles all low level USB protocol in hardware. We use USB bulk transfers. To the CPLD the FX2LP appears as a FIFO sink of AER data. The 8051 core configures the FIFOs and the interface, but is not in the data path.

The 8051 core is also used to control the bias generator on the chip, start and stop event acquisition, signal a time-stamp reset and to load the CPLD internal configuration shift register.

A.4 Host side

The firmware of the cDVSTest PCB is designed to be interfaced with the jAER software package [81]. jAER is an extensive open-source Java software package designed to record, visualize and process event streams from event-based hardware. Event acquisition is handled by a USB driver developed by Thesycon [142] and uses overlapped I/O.

The jAER software package works with 32 bit timestamps. However, recording and sending 32 bit timestamps from the device to the host computer is very inefficient, as the most significant bits change only very rarely. To save bandwidth, the device sends 14 bit timestamps to the host computer. On the host computer these timestamps are unwrapped to 32 bits. The host computer adds the "wrap-add" to every time-stamp it receives from the device. To hold this wrap-add up-to-date, the device sends a special event every time the 14-bit counter overflows and therefore the wrap-add should be increased by $2^{14} = 16384$. Another class of special events tells the host computer the reset the wrap-add to zero.

Appendix B

Measurement Setup

Spectral sensitivities of the buried double junctions built in $1.6\mu m$ and the $0.5\mu m$ process technology have been measured with a Mitutoyo Microscope light source and an Optometrics MC1-03 monochromator with a $300\mu m$ slit, which gives roughly 2nm resolution [143].

The pattern detection circuit of DollBrain1 was characterized by pointing the DollBrain1 PCB equipped with a lens (illustrated in Fig. 3.6) at an uncalibrated LCD screen and stimulating it with patterns generated and varied in Matlab.

For characterization of the change detection pixels, the chips were illuminated using high-power blue (450nm, 20nm bandwidth) and red (630nm, 15nm bandwidth) Seoul Z-Power LEDs, mixed together with a dual branch light guide (Dolan-Jenner EE836). Varying the ratio of currents through the red LED and the blue LED changes the mean wavelength of the emitted light. Fig. B.1 shows a picture of the LED driver stage, the schematic is shown in Fig. B.2. Irradiance at the output of the light guide is $7.45W/m^2$ for the blue LED and $12.18W/m^2$ for the red LED when the LEDs are driven with 200mA current. This corresponds to roughly 0.5klux and 2.2klux illuminance respectively.

The LED driver is powered with a HP E3610A power supply. LED DC currents are set with a Keithley K230 voltage source, the currents are modulated with HP 33120A function generators. A Tektronix TDS6054B oscilloscope was used to capture voltage traces. Light intensities were measured with a Tektronix J17 photometer using the J1812 irradiance head. Noise voltage spectra were measured with a Stanford Research SR780 spectrum analyzer. Fig. B.3 shows the lab setup.

To characterize the ColTmpDiff chip, it was set up in PotBox, built at the Institute of Neuroinformatics. This PotBox, shown in Fig. B.4, has ten potentiometers to set chip bias voltages. The ColTmpDiff chip needs 18 bias voltages. One bias voltage was set with a Keithley K236 source measure unit, the remaining seven biases were controlled by additional potentiometers.

cDVSTest10 and cDVSTest30 were controlled using the cDVSTest PCB described in Appendix A.

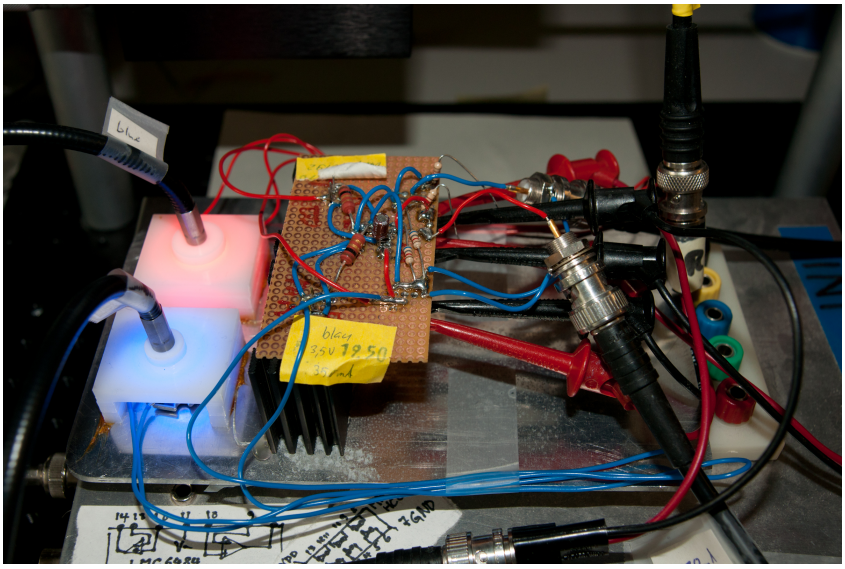


Figure B.1: LED driver stage

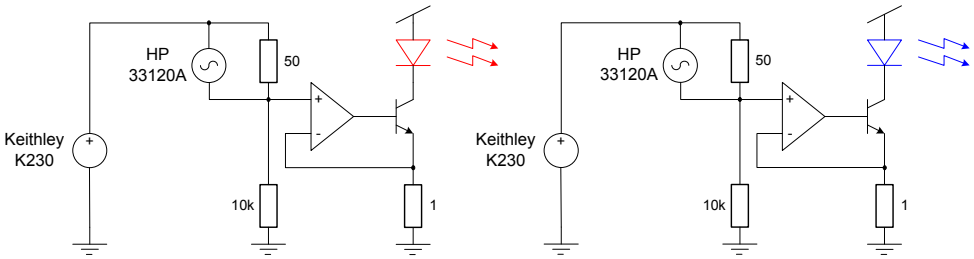


Figure B.2: Schematic of LED driver stage

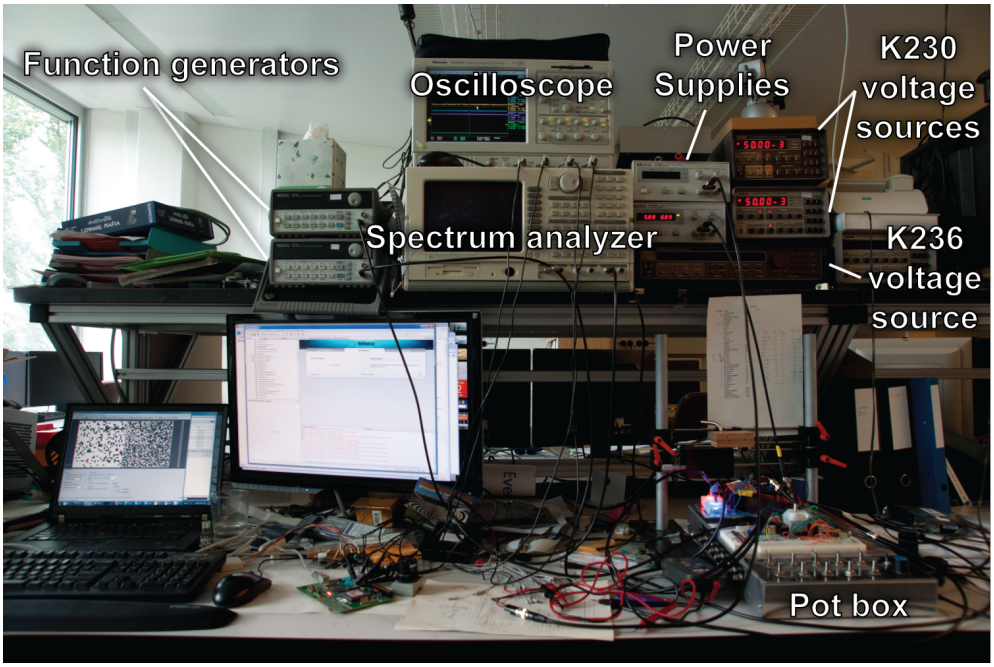


Figure B.3: Lab setup. The function generators are used to modulate the LEDs, the Keithley K230 voltage sources set the DC current through the LEDs. The Keithley K236 is used for computer controlled parameter changing when using the PotBox. One of the power supplies is used to power the PotBox, the other one powers the LED driver.

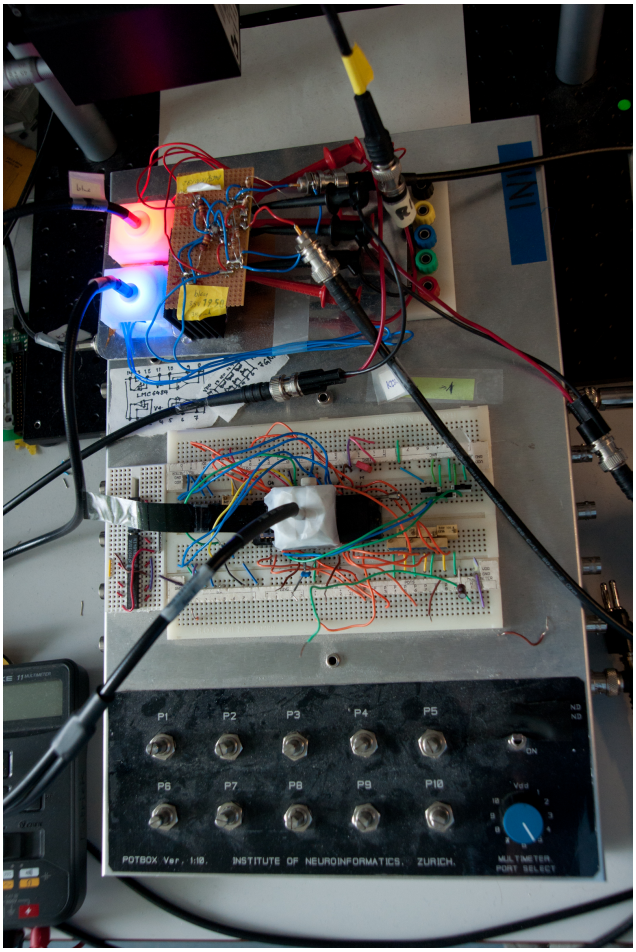


Figure B.4: Pot box with ColTmpDiff chip and LED driver stage.

Appendix C

Short LogSpice Tutorial

C.1 Introduction

LogSpice has been developed by John Arthur and Paul Merolla at the Lab of Kwabena Boahen at Standford University. It is a set of Matlab functions that can be used to simulate the behavior of asynchronous digital circuits in the presence of signal delays. It consists of a set of text files, which are loaded into Matlab and processed there.

LogSpice is a very simple hardware description language for digital systems. The specialty is that it introduces arbitrary delays between signals (for each simulation run drawn from a uniform distribution), and thus tests whether a design is delay insensitive.

In a LogSpice simulation, all signals are digital and know only the logic levels '0' and '1', thus the signals make instantaneous transistions. RC-delays and slowly rising edges can not be simulated with this tool.

A more thorough description of LogSpice, the available Matlab functions and all the commands can be found in the Readme file of LogSpice.

C.2 Basic Components, Sub-circuits and Commands

LogSpice knows three basic components: LPU, LPD and LPULSE. Table C.1 shows detailed explanation and syntax.

Everything is constructed from combinations of these basic components. Each signal must be either defined by one LPULSE statement or have exactly one LPU and one LPD. Signals can negated by placing a tilde in front of the signal name. The condition can be an equation of several signals, enclosed in curly brackets. Equations make use of `|` and `&` as logical OR and AND operators. The following code example defines that *out* is the logical AND operation of *in1* and *in2*.

```
LDUout out {(in1 & in2)} LPU
LPDout out {(~in1 | ~in2)} LPD
```

Component	Syntax	Description
Logical Pull-Up (LPU)	`LPUname signal condition LPU`	If *condition* is true, *signal* becomes true.
Logical Pull-Down (LPD)	`LPDname signal condition LPD`	If *condition* is true, *signal* becomes false.
Input Waveform (LPULSE)	`xname signal t1 v1 t2 v2 ... LPULSE`	*Signal* takes the value v_i (0 or 1) at the time t_i.

Table C.1: Basic LogSpice Components

For each simulation run, LogSpice will introduce a delay between the input and the output signals of the LPU and LPD components. The range of delays can be specified with the following code:

```
% set delay variance (drawn from a uniform dist) for all variables
.delay delayMin delayMax

% sets specific delay variance between varNameA and varNameB
.delaySpec (varNameA) (varNameB) delayMin delayMax
% varNameA and varNameB can be wildcard references.
```

In a truly delay-insensitive design, any combination of delay must not impede the system from working. A single `.delay` statement valid for all signals would be sufficient. In reality, the class of entirely delay-insensitive circuits is very limited [129, 144]. In our design, the delay assumption is that the delay from long range signals to the two state variables in a state machine is similar, as well as the delay from the state variables to an output of the state machine is similar. Therefore it is necessary to have separate `.delay` statements for state variables and for long range signals.

LogSpice allows to build sub-circuits from the LPU and LPD components.

```
.subckt AND2 out in1 in2
LDUout out {(in1 & in2)} LPU
LPDout out {(~in1 | ~in2)} LPD
.ends
```

The code to instantiate a sub-circuit is as follows:

```
xexample ExampleOut ExampleIn1 ExampleIn2 AND2
```

Sub-circuits themselves can also instantiate other sub-circuits to form hierarchical designs.

Additional LogSpice commands allow to include other files describing subcircuits, set the number of simulation runs, set the length of the simulation and define the signals to be plotted.

```
% includes dependent files (which in turn may also include other
% dependents)
.include fileName

% sets number of trials to run
.trials trialNum

% sets timestep number
.tran timeSteps

% flags the variable names that should plotted; also uses wildcard
% references.  Plot can take any number of parameters (indicated
% by ...).
% The order of the parameters controlls the ploting order
%(last parameter is ploted on top)
.plot (var1) (var2) ...
% different plot statements can appear in multiple windows
% (depending on the plot options)
.plot (var3) (var4) ...

.options plotOption
% option can be:
% plotAll (plots all the variables),
% plotWild (plots the all variables that have been flagged
%   in a single window)
% plotWildSep (flagged variables in each plot statement
%   are shown in separate windows).
```

C.3 Using LogSpice

To use LogSpice, you need to load the text file describing the simulation setup into Matlab. Then there is a single command which simulates the described circuit as many times as indicated with the `.trials` command in the input file.

```
fileIn = 'testbench.txt';
[xt delayMat sOut] = prsBatchRunf(fileIn);
```

The output of LogSpice are time-line plots like in Fig. 7.3. LogSpice also compares the sequence of signal transitions of each simulation trial to the other trials and plots a similarity measure. This similarity measure is a useful indication which

trial differs most from other trials, but it does not necessarily mean that a circuit is malfunctioning if the sequence of signal transitions is not equal from trial to trial.

Additionally, LogSpice plots in the command window if there is any interference for a signal at some time point. Interference means that for the reported signal, the pull-down and the pull-up path are active simultaneously at the given time point.

LogSpice does not tell the user whether a circuit is working correctly, the user has to examine the time-line plots and check whether the sequence of signal transmissions is correct or not.

Bibliography

[1] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in cognitive sciences*, vol. 5, pp. 62–71, Feb. 2001. 1.1

[2] C. Mead, *Analog VLSI and neural systems*. Boston, MA: Addison-Wesley, 1989. 1.1

[3] M. Maher, S. Deweerth, M. Mahowald, and C. Mead, "Implementing neural architectures using analog VLSI circuits," *IEEE Trans. Circuits Syst.*, vol. 36, no. 5, pp. 643–652, May 1989. 1.1

[4] S. Still, K. Hepp, and R. Douglas, "Neuromorphic walking gait control," *Neural Networks, IEEE Transactions on*, vol. 17, no. 2, pp. 496 –508, Mar. 2006. 1.1

[5] Y. Wang and S.-C. Liu, "Input evoked nonlinearities in silicon dendritic circuits," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2009, pp. 2894 –2897. 1.1

[6] E. Neftci, E. Chicca, G. Indiveri, M. Cook, and R. Douglas, "State-dependent sensory processing in networks of VLSI spiking neurons," in *Proc. IEEE Int. Symp. Circuits Syst.* IEEE, 2010, pp. 2789–2792. 1.1

[7] C. Mead and M. A. Mahowald, "A silicon model of early visual processing," *Neural Networks*, vol. 1, no. 1, pp. 91–97, 1988. 1.1, 1.4, 1.4.2

[8] Facets project. [Online]. Available: http://facets.kip.uni-heidelberg.de/ 1.1

[9] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst*, Paris, France, 2010, pp. 1947–1950. 1.1

[10] BrainScaleS project. [Online]. Available: http://brainscales.kip.uni-heidelberg.de/ 1.1

[11] K. Boahen. Neurogrid project. [Online]. Available: http://www.stanford.edu/group/brainsinsilicon/neurogrid.html 1.1, 7.4

[12] X. Jin, M. Lujan, L. A. Plana, S. Davies, S. Temple, and S. Furber, "Modeling spiking neural networks on SpiNNaker," *Computing in Science & Engineering*, vol. 12, pp. 91–97, Sep. 2010. 1.1

[13] SpiNNaker project. [Online]. Available: http://apt.cs.man.ac.uk/projects/SpiNNaker/ 1.1

[14] T. Delbruck and S. Liu, "A silicon visual system as a model animal," *Vision Research*, vol. 44, pp. 2083–2089, 2004. 1.1

[15] J. Dethier, V. Gilja, P. Nuyujukian, S. A. Elassaad, K. V. Shenoy, and K. Boahen, "Spiking neural network decoder for brain-machine interfaces," in *IEEE EMBS Conference on Neural Engineering*. IEEE, 2011. 1.1

[16] R. Jung, E. Brauer, and J. Abbas, "Real-time interaction between a neuromorphic electronic circuit and the spinal cord," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 9, no. 3, pp. 319 –326, Sep. 2001. 1.1

[17] R. Vogelstein, F. Tenore, L. Guevremont, R. Etienne-Cummings, and V. Mushahwar, "A silicon central pattern generator controls locomotion in vivo," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 2, no. 3, pp. 212 –222, Sep. 2008. 1.1

[18] X. Arreguit, A. van Schaik, F. Bauduin, M. Bidiville, and E. Raeber, "A CMOS motion detector system for pointing devices," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1916–1921, Dec. 1996. 1.1

[19] F. Zhengming, T. Delbruck, P. Lichtsteiner, and E. Culurciello, "An address-event fall detector for assisted living applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, pp. 88–96, Jun. 2008. 1.1, 1.5

[20] M. Litzenberger, B. Kohn, G. Gritsch, N. Donath, C. Posch, N. Belbachir, and H. Garn, "Vehicle counting with an embedded traffic data system using an optical transient sensor," in *Intelligent Transportation Systems Conference. IEEE*, Sep. 2007, pp. 36–40. 1.1, 1.5

[21] Z. Fu, C. Joshi, and A. Titus, "CMOS-based colour-change pH measurement system," *Electronics Letters*, vol. 45, no. 22, pp. 1138 –1140, Oct. 2009. 1.1

[22] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, no. 0, 2011. 1.1

[23] T. J. Koickal, A. Hamilton, S. L. Tan, J. A. Covington, J. W. Gardner, and T. C. Pearce, "Analog VLSI circuit implementation of an adaptive neuromorphic olfaction chip," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 1, pp. 60 –73, Jan. 2007. 1.1

[24] R. Lyon and C. Mead, "A CMOS VLSI cochlea," in *Acoustics, Speech, and Signal Processing, International Conference on*, vol. 4, Apr. 1988, pp. 2172 –2175. 1.1

[25] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 1, pp. 48 –59, Jan. 2007. 1.1

[26] M. Sivilotti, "Wiring considerations in analog VLSI systems with application to field-programmable networks," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1991. 1.1.1

[27] R. Serrano-Gotarredona *et al.*, "CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory-Processing- Learning-Actuating System for High-Speed Visual Object Recognition and Tracking," *IEEE Trans. Neural Netw.*, vol. 20, pp. 1417–1438, 2009. 1.1.1, 1.5

[28] F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miro, S. Vicente, G. Jimenez, and A. Civit, "AER tools for communications and debugging," in *Proc. IEEE Int. Symp. Circuits Syst*, 2006, pp. 3253–3256. 1.1.1

[29] R. Paz-Vicente, A. Linares-Barranco, D. Cascado, M. Rodriguez, G. Jimenez, A. Civit, and J. Sevillano, "PCI-AER interface for neuro-inspired spiking systems," in *Proc. IEEE Int. Symp. Circuits Syst*, 2006, pp. 3161–3164. 1.1.1

[30] R. Berner, T. Delbruck, A. Linares-Barranco, and A. Civit, "A 5 Meps $100 USB2.0 Address-Event Monitor-Sequencer Interface," in *Proc. IEEE Int. Symp. Circuits Syst*, 2007, pp. 2451–2454. 1.1.1, A, A.2.4

[31] K. Boahen, "A burst-mode word-serial address-event link-I: transmitter design," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 7, pp. 1269 – 1280, Jul. 2004. 1.1.1, 7.2, 7.3, 7.4, 7.10.1, 7.10.3, 7.11, 7.13, 9.3

[32] ——, "A burst-mode word-serial address-event link-II: receiver design," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 7, pp. 1281 – 1291, Jul. 2004. 1.1.1

[33] ——, "A burst-mode word-serial address-event link-III: analysis and test results," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 7, pp. 1292 – 1300, Jul. 2004. 1.1.1, 7.12, 9.3

[34] P. A. Merolla, J. V. Arthur, B. E. Shi, and K. A. Boahen, "Expandable networks for neuromorphic chips," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 2, pp. 301 –311, Feb. 2007. 1.1.1

[35] L. Miro-Amarante, A. Jimenez, A. Linares-Barranco, F. Gomez-Rodriguez, R. Paz, G. Jimenez, A. Civit, and R. Serrano-Gotarredona, "A LVDS serial AER link," in *Electronics, Circuits and Systems. 13th IEEE International Conference on*, Dec. 2006, pp. 938 –941. 1.1.1

[36] D. Fasnacht, A. Whatley, and G. Indiveri, "A serial communication infrastructure for multi-chip address event systems," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2008, pp. 648 –651. 1.1.1

[37] H. Berge and P. Hafliger, "High-speed serial AER on FPGA," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2007, pp. 857 –860. 1.1.1

[38] C. Zamarreno-Ramos, R. Serrano-Gotarredona, T. Serrano-Gotarredona, and B. Linares-Barranco, "LVDS interface for AER links with burst mode operation capability," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2008, pp. 644 –647. 1.1.1

[39] P. Lichtsteiner, "An AER temporal contrast vision sensor," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2006, DISS. ETH No. 16898. 1.2, 1.3, 1.4, 1.5, 1.6, 1.5, 1.7, 5.3, 6.2, 7.2, 7.12, 8.4

[40] B. Wandell, *Foundations of Vision.* Sunderland, MA, USA: Sinauer Associates, 1995. 1.2.2

[41] T. Gollisch and M. Meister, "Eye smarter than scientists believed: Neural computations in circuits of the retina," *Neuron*, vol. 65, pp. 150–164, Dec. 2009. 1.2.3

[42] C. Chao, H. Tu, K. Chou, P. Chou, F. Hsueh, W. Wei, R. Lin, and B. Hseih, "Crosstalk metrics and the characterization of 1.1μm-pixel CIS," in *International Image Sensor Workshop*, Jun. 2011. 1.3.1

[43] E. R. Fossum, "Active pixel sensors: Are CCD's dinosaurs?" in *Proc. SPIE Vol. 1900, Charge-Coupled Devices and Solid State Optical Sensors III*, M. M. Blouke, Ed. SPIE, 1993, pp. 2–14. 1.3.2, 1.4.1, 2.1

[44] R. Nixon, S. Kemeny, B. Pain, C. Staller, and E. Fossum, "256×256 CMOS active pixel sensor camera-on-a-chip," *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 2046–2050, Dec. 1996. 1.3.2

[45] 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. 1.3.2, 8.3.1, 9.2

[46] C. Lotto, P. Seitz, and T. Baechler, "A sub-electron readout noise CMOS image sensor with pixel-level open-loop voltage amplification," in *Solid-State Circuits Conference Digest of Technical Papers. ISSCC 2011. IEEE International*, Feb. 2011, pp. 402 –404. 1.3.2

[47] T. Delbruck, "Silicon retina with correlation-based, velocity-tuned pixels," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 529–541, May 1993. 1.4

[48] A. Stocker, "Analog VLSI focal-plane array with dynamic connections for the estimation of piecewise-smooth optical flow," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 5, pp. 963 – 973, May 2004. 1.4

[49] T. Delbruck, "Silicon retina for autofocus," in *Proc. IEEE Int. Symp. Circuits Syst*, vol. 4, 2000, pp. 393–396. 1.4

[50] V. Chan, C. Jin, and A. van Schaik, "An address-event vision sensor for multiple transient object detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 1, no. 4, pp. 278–288, Dec. 2007. 1.4

[51] M. A. Mahowald, "Silicon retina with adaptive photoreceptors," *Proc. Int. Soc. Optical Engineers-Visual Information Processing: From Neurons to Chips*, vol. 1473, pp. 52–58, Apr. 1991. 1.4

[52] M. Mahowald, "Analog VLSI chip for stereocorrespondence," in *Proc. IEEE Int. Symp. Circuits Syst*, vol. 6, May 1994, pp. 347–350. 1.4

[53] K. Zaghloul and K. Boahen, "Optic nerve signals in a neuromorphic chip," *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 657–675, Dec. 2004. 1.4

[54] 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. 1.4.1, 1.6, 2.1

[55] J. Olsson and P. Hafliger, "Mismatch reduction with relative reset in integrate-and-fire photo-pixel array," in *IEEE Biomedical Circuits and Systems Conference*, Nov. 2008, pp. 277 –280. 1.4.1

[56] X. Qi, X. Guo, and J. G. Harris, "A time-to-first spike CMOS imager," in *Proc. IEEE Int. Symp. Circuits Syst*, 2004, pp. 23–26. 1.4.1, 2.1

[57] C. Shoushun and A. Bermak, "Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 346 –357, Mar. 2007. 1.4.1, 2.1

[58] 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. 1.4.1, 2.1, 3.3.3

[59] C. Shoushun and A. Bermak, "A second generation time-to-first-spike pixel with asynchronous self power-off," in *Proc. IEEE Int. Symp. Circuits Syst*, 2008. 1.4.1, 2.1

[60] S. Kameda and T. Yagi, "An analog VLSI chip emulating sustained and transient response channels of the vertebrate retina," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1405 – 1412, Sep. 2003. 1.4.2

[61] R. Takami, K. Shimonomura, S. Kameda, and T. Yagi, "An image preprocessing system employing neuromorphic 100 × 100 pixel silicon retina," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2005, pp. 2771–2774. 1.4.2

[62] K. Shimonomura and T. Yagi, "A silicon retina system for color constancy," in *World Automation Congress (WAC)*, Sep. 2010, pp. 1 –5. 1.4.2, 1.6

[63] P. F. Ruedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P. Y. Burgi, S. Gyger, and P. Nussbaum, "A 128 × 128 pixel 120-dB dynamic-range vision-sensor chip for image contrast and orientation extraction," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2325–2333, Dec. 2003. 1.4.2

[64] J. Leñero Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A five-decade dynamic-range ambient-light-independent calibrated signed-spatial-contrast AER retina with 0.1-ms latency and optional time-to-first-spike mode," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 10, pp. 2632–2643, Oct. 2010. 1.4.2

[65] K. Boahen and A. Andreou, "A contrast-sensitive retina with reciprocal synapses," in *Advances in Neural Information Processing*, J. E. Moody, Ed. San Mateo, CA: Morgan Kaufmann, 1992, pp. 764–772. 1.4.2

[66] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120dB 15us latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, pp. 566–576, Feb. 2008. 1.4.2, 1.4.3, 1.5, 1.7, 4, 4.4, 5.3, 6, 6.1.1, 6.1.2, 7.10.3

[67] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 db dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259 –275, Jan. 2011. 1.4.2, 1.4.3, 2.1, 7.13, 8.3, 9.2, 9.2, 9.3, 9.4

[68] A. Lopich and P. Dudek, "An 80 × 80 general-purpose digital vision chip in $0.18\mu m$ CMOS technology," in *Proc. IEEE Int. Symp. Circuits Syst*, Jun. 2010, pp. 4257 –4260. 1.4.2, 1.4.4

[69] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "Temporal change threshold detection imager," in *Solid-State Circuits Conference Digest of Technical Papers. ISSCC 2005. IEEE International*, Feb. 2005, pp. 362–363. 1.4.3

[70] J. Kramer, "An ON/OFF transient imager with event-driven, asynchronous readout," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2002. 1.4.3

[71] ——, "An integrated optical transient sensor," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 9, pp. 612–628, Sep. 2002. 1.4.3

[72] J. Leñero Bardallo, T. Serrano-Gottarredona, and B. Linares-Barranco, "A 3.6μs latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, Jun. 2011. 1.4.3

[73] P. Dudek and P. Hicks, "A general-purpose processor-per-pixel analog SIMD vision chip," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 1, pp. 13 – 20, Jan. 2005. 1.4.4

[74] P. Dudek, "Implementation of SIMD vision chip with 128 × 128 array of analogue processing elements," in *Proc. IEEE Int. Symp. Circuits Syst*, vol. 6, May 2005, pp. 5806 – 5809. 1.4.4

[75] P. Dudek and S. Carey, "General-purpose 128 × 128 SIMD processor array with integrated image sensor," *Electronics Letters*, vol. 42, no. 12, pp. 678 – 679, Jun. 2006. 1.4.4

[76] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 5, pp. 416–434, May 2000. 1.5, 3.3.3, 4.1.1, 6.1.1, 7.2, 7.10.3, 7.10.5

[77] T. Delbruck and P. Lichtsteiner, "Fully programmable bias current generator with 24 bit resolution per bias," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2006, pp. 2849–2852. 1.5, 6.2.1

[78] C. Shoushun, B. Martini, and E. Culurciello, "A bio-inspired event-based size and position invariant human posture recognition algorithm," in *Proc. IEEE Int. Symp. Circuits Syst*, 2009, pp. 775–778. 1.5

[79] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst*, 2009, pp. 781–784. 1.5

[80] T. Delbruck and R. Berner, "Temporal contrast AER pixel with 0.3%-contrast event threshold," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2010, pp. 2442–2445. 1.5, 4.2, 2

[81] jAER open source project. [Online]. Available: http://jaer.wiki.sourceforge.net 1.5, 3.4, A.4

[82] R. W. Rodieck, *The First Steps in Seeing*. Sunderland, MA, USA: Sinauer Associates, 1998. 1.6, 2.4

[83] J. Olsson and P. Hafliger, "Two color asynchronous event photo pixel," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2008, pp. 2146–2149. 1.6, 2.3, 9.1, 9.4

[84] Z. Fu and A. H. Titus, "CMOS neuromorphic optical sensor chip with color change-intensity change disambiguation," *IEEE Sensors J.*, vol. 9, no. 6, pp. 689–696, Jun. 2009. 1.6, 2.3, 4.1

[85] T. Delbruck and C. Mead, "Adaptive photoreceptor with wide dynamic range," in *Proc. IEEE Int. Symp. Circuits Syst*, vol. 4, May 1994, pp. 339–342. 1.6

[86] R. Berner, P. Lichtsteiner, and T. Delbruck, "Self-timed vertacolor dichromatic vision sensor for low power pattern detection," in *Proc. IEEE Int. Symp. Circuits Syst*, 2008, pp. 1032 – 1035. 1.7, 2.3

[87] R. Berner and T. Delbruck, "Event-based color change pixel in standard CMOS," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2010, pp. 349 –352. 1.7

[88] ——, "Event-based pixel sensitive to changes of color and brightness," *IEEE Trans. Circuits Syst. I*, vol. 58, pp. 1581–1590, Jul. 2011. 1.7

[89] E. Vittoz and J. Fellrath, "CMOS analog integrated circuits based on weak inversion operations," *IEEE J. Solid-State Circuits*, vol. 12, no. 3, pp. 224 – 231, Jun. 1977. 1.8

[90] E. Vittoz, "Weak inversion for ultra low-power and very low-voltage circuits," in *Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian*, Nov. 2009, pp. 129–132. 1.8

[91] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras.* Boca Raton, FL, USA: CRC Press, Inc., 2005. 2

[92] S. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI: Circuits and Principles.* Cambridge, Massachusetts: The MIT Press, 2002. 2, 2.3, 4.1.4

[93] D. J. Roulston, *An Introduction to the Physics of Semiconductor Devices.* New York, USA: Oxford University Press, 1998. 2

[94] A. Grove, *Physics and Technology of Semiconductor Devices.* Hoboken, USA: Wiley, 1967. 2, 2.4.1

[95] M. A. Green and M. J. Keevers, "Optical properties of intrinsic silicon at 300K," *Progress in Photovoltaics: Research and Applications*, vol. 3, pp. 189 – 192, 1995. 2.1, 2.3

[96] M. Law, E. Solley, M. Liang, and D. Burk, "Self-consistent model of minority-carrier lifetime, diffusion length, and mobility," *Electron Device Letters, IEEE*, vol. 12, no. 8, pp. 401–403, Aug. 1991. 2.1.1, 2.4.1

[97] D. J. Roulston, N. D. Arora, and S. G. Chamberlain, "Modeling and measurement of minority-carrier lifetime versus doping in diffused layers of n^+p silicon diodes," *IEEE Transactions on Electron Devices*, vol. 29, no. 2, pp. 284–291, Feb. 1982. 2.1.1, 2.4.1

[98] Color filter array, wikipedia article. [Online]. Available: http://en.wikipedia.org/wiki/Color_filter_array 2.6

[99] Europractice Multi Project Wafer service. [Online]. Available: http://www.europractice-ic.com 2.1, 6.2, 8.4

[100] H. Honda, Y. Iida, Y. Egawa, and H. Seki, "A color CMOS imager with 4×4 white-RGB color filter array for increased low-illumination signal-to-noise ratio," *Electron Devices, IEEE Transactions on*, vol. 56, no. 11, pp. 2398 –2402, Nov. 2009. 1

[101] W. N. Carr, "Multi-spectrum photodiode devices," U.S. Patent 4 238 760, Dec. 9, 1980. 2.3

[102] T. Yoshikawa, Z. Tani, A. Aso, and H. Kawanabe, "Apparatus for sensing the wavelength and intensity of light," U.S. Patent 4 309 604, Jan. 5, 1982. 2.3

[103] R. F. Wolffenbuttel, "Color filters integrated with the detector in silicon," *IEEE Electron Device Lett.*, vol. 8, pp. 13–15, Jan. 1987. 2.3

[104] G. de Graaf and R. F. Wolffenbuttel, "Light-to-frequency converter using integrating mode photodiodes," *IEEE Trans. Instrum. Meas.*, vol. 46, pp. 933–936, Aug. 1997. 2.3

[105] C. Yuan, X. Zhihai, and F. Huajun, "A CMOS image sensor for monochromatic spectrum imaging," in *Imaging Systems and Techniques*, May 2009, pp. 6–10. 2.3

[106] D. Fasnacht and T. Delbruck, "Dichromatic spectral measurement circuit in vanilla CMOS," in *Proc. IEEE Int. Symp. Circuits Syst*, 2007, pp. 3091–3094. 2.3, 2.7, 2.4.2, 3.3.1, 3.2

[107] Foveon Inc. [Online]. Available: http://www.foveon.com/ 2.3

[108] L. Paulson, "Will new chip revolutionize digital photography?" *Computer*, vol. 35, no. 5, pp. 25 –26, May 2002. 2.3

[109] G. Gilder, *The Silicon Eye: How a Silicon Valley Company Aims to Make All Current Computers, Cameras, and Cell Phones Obsolete.* New York: W.W. Norton & Company, 2005. 2.3

[110] I. Brouk and Y. Nemirovsky, "Dimensional effects in CMOS photodiodes," *Solid-State Electronics*, vol. 46, no. 1, pp. 19 – 28, Jan. 2002. 2.8(a)

[111] M. Sedjil, G. Lu, M. B. Chouikha, and A. Alexandre, "Modeling of BDJ and BTJ structures for color detection," *Proc. SPIE, Design, Test, and Microfabrication of MEMS and MOEMS*, vol. 3680, pp. 388–397, Mar. 1999. 2.4.1

[112] T. Delbruck, T. Koch, R. Berner, and H. Hermansky, "Fully integrated $500\mu W$ speech detection wake-up circuit," in *Proc. IEEE Int. Symp. Circuits Syst*, Jun. 2010, pp. 2015 –2018. 3

[113] R. Etienne-Cummings, P. Pouliquen, and M. A. Lewis, "A vision chip for color segmentation and pattern matching," *Eurasip Journal on Applied Signal Processing*, vol. 2003, no. 7, pp. 703–712, 2003. 3.1

[114] Y. Hori and T. Kuroda, "A $0.79mm^2$ $29mW$ real-time face detection core," *IEEE J. Solid-State Circuits*, vol. 42, pp. 790–797, 2007. 3.1

[115] E. Angelopoulou, "Understanding the color of human skin," *2001 SPIE Conference on Human Vision and Electronic Imaging*, pp. 243–251, 2001. 3.2

[116] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. 511–518. 3.2

[117] ——, "Robust real-time face detection," in *Computer Vision. Proceedings of the Eighth IEEE International Conference on*, vol. 2, 2001, pp. 747 –747. 3.2

[118] T. Delbruck and A. van Schaik, "Bias current generators with wide dynamic range," *Analog Integrated Circuits and Signal Processing*, vol. 43, pp. 247–268, Jun. 2005. 3.4, 6.2.1

[119] C. Posch, D. Matolin, and R. Wohlgenannt, "A two-stage capacitive-feedback differencing amplifier for temporal contrast IR sensors," in *Electronics, Circuits and Systems. 14th IEEE International Conference on*, May 2007, pp. 1071 –1074. 4.1.1, 6.1.1, 6.3, 2

[120] T. Serrano-Gotarredona, B. Linares-Barranco, and A. Andreou, "Very wide range tunable cmos/bipolar current mirrors with voltage clamped input," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 46, no. 11, pp. 1398 –1407, Nov. 1999. 4.1.2

[121] P. Merkli, "Message-Passing Algorithms and Analog Electronic Circuits," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2005, DISS. ETH No. 15942. 4.1.3

[122] B. Linares-Barranco and T. Serrano-Gotarredona, "On the design and characterization of femtoampere current-mode circuits," *IEEE J. Solid-State Circuits*, vol. 38, no. 8, pp. 1353 – 1363, Aug. 2003. 4.3, 4.1.3, 4.4

[123] D. A. Johns and K. Martin, *Analog Integrated Circuit Design.* New York: John Wiley & Sons, 1997. 5.2, 8.3.2

[124] T. Delbruck, "Investigations of analog VLSI visual transduction and motion processing," Ph.D. dissertation, California Institute of Technology, Departement of Computation and Neural Systems, Pasadena, CA, 1993. 5.3

[125] R. Sarpeshkar, T. Delbruck, and C. Mead, "White-noise in MOS-transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 9, pp. 23–29, Nov. 1993. 5.3

[126] T. Delbruck, R. Berner, P. Lichtsteiner, and C. Dualibe, "32-bit configurable bias current generator with sub-off-current capability," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2010, pp. 1647 –1650. 6.1.1, 6.2.1

[127] T. Delbruck and A. van Schaik, "Bias current generators with wide dynamic range," in *Proc. IEEE Int. Symp. Circuits Syst*, vol. 1, May 2004, pp. 337–340. 6.2.1

[128] A. Rose, *Vision: Human and Electronic.* New York: Plenum Press, 1973. 6.3

[129] A. J. Martin and M. Nyström, "Asynchronous techniques for system-on-chip design," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1089 –1120, Jun. 2006. 7.4, 7.10, C.2

[130] 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. 8.3.1

[131] N. Ricquier and B. Dierickx, "Pixel structure with logarithmic response for intelligent and flexible imager architectures," in *Solid State Device Research Conference, 22nd European*, Sep. 1992, pp. 631 –634. 8.3.1

[132] 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. 8.3.1

[133] N. Ricquer and B. Dierickx, "Active pixel CMOS sensor with on chip nonuniformity correction," in *Proc. IEEE Workshop CCD AIS*, 1995, pp. 20–21. 8.3.1

[134] Y. Wong and P. Abshire, "A 144×144 current-mode image sensor with self-adapting mismatch reduction," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 8, pp. 1687–1697, Aug. 2007. 8.3.1

[135] Y. Ni and K. Matou, "A CMOS log image sensor with on-chip FPN compensation," in *Proceedings of the 27th European Solid-State Circuits Conference*, Sep. 2001, pp. 101–104. 8.3.1

[136] L.-W. Lai, C.-H. Lai, and Y.-C. King, "A novel logarithmic response CMOS image sensor with high output voltage swing and in-pixel fixed-pattern noise reduction," *IEEE Sensors J.*, vol. 4, no. 1, pp. 122 – 126, Feb. 2004. 8.3.1

[137] 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. 8.3.1

[138] B. Choubey, S. Aoyoma, S. Otim, D. Joseph, and S. Collins, "An electronic-calibration scheme for logarithmic CMOS pixels," *IEEE Sensors J.*, vol. 6, no. 4, pp. 950–956, Aug. 2006. 8.3.1, 8.3.2, 9.2

[139] S. Otim, B. Choubey, D. Joseph, and S. Collins, "Characterization and simple fixed pattern noise correction in wide dynamic range logarithmic imagers," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1910–1916, Oct. 2007. 8.3.1

[140] H.-Y. Cheng, B. Choubey, and S. Collins, "An integrating wide dynamic-range image sensor with a logarithmic response," *IEEE Trans. Electron Devices*, vol. 56, no. 11, pp. 2423–2428, Nov. 2009. 8.3.1

[141] R. Berner, "Highspeed USB2.0 AER Interfaces," Master's thesis, ETH Zurich, Switzerland and University of Seville, Spain, 2006. A, A.2, A.2.4

[142] Thesycon GmbH. [Online]. Available: http://www.thesycon.de A.4

[143] Optometrics Monochromator. [Online]. Available: http://www.optometrics.com/App_Themes/optometrics/pdfs/Monochromator10_8_10.pdf B

[144] A. J. Martin, "The limitations to delay-insensitivity in asynchronous circuits," in *Proc. 6th MIT Conf. Advanced Research in VLSI*, W. Dally, Ed., 1990, pp. 263 –278. C.2

Curriculum vitae

Raphael Berner
Born on May 14., 1980 in Aarau, Switzerland
Citizen of Schafisheim, AG, Switzerland

Education

1987 - 1992	Primary school in Erlinsbach, AG
1992 - 1996	Secondary school Bezirksschule Aarau, AG
1996 - 2000	Matura Typus C Alte Kantonsschule Aarau
2000 - 2006	Master of Science in Electrical Engineering and Information Technology, ETH Zurich
Jan. - Jun. 2004	Erasmus Semester at Lunds Tekniska Högskola, Lund, Sweden
Sep. 2005 - Apr. 2006	Master project at the University of Sevilla, Spain
Since Feb. 2007	PhD student at the Institute of Neuroinformatics, University of Zurich and ETH Zurich

Professional Experience

Feb. - Mar. 2001	Internship at Precision Implants AG, Aarau, Switzerland.
Mar. - Jun. 2002	Teaching assistant for “Netzwerke und Schaltungen 2”, taught by Professor H. Jäckel.
Aug. - Dez. 2003	Internship at Studer Professional Audio AG, Regensdorf, Switzerland.
Oct. 2004 - Jan. 2005	Teaching assistant for “Elektrotechnik für Informatiker”, taught by Professor R. Vahldieck.

Scientific Publications

A 5 Meps $100 USB2.0 Address-Event Monitor-Sequencer Interface. Berner, R.; Delbruck, T.; Civit-Balcells, A.; Linares-Barranco, A.; IEEE International Symposium on Circuits and Systems, 2007, Pages: 2451 - 2454.

Self-timed vertacolor dichromatic vision sensor for low power pattern detection. Berner, R.; Lichtsteiner, P.; Delbruck, T.; IEEE International Symposium on Circuits and Systems, 2008, Pages: 1032 - 1035.

Event-based color change pixel in standard CMOS. Berner, R.; Delbruck, T.; IEEE International Symposium on Circuits and Systems, 2010, Pages: 349 - 352.

Event-Based Pixel Sensitive to Changes of Color and Brightness. Berner, R.; Delbruck, T.; IEEE Transactions on Circuits and Systems I: Regular Papers, Volume 58, Issue 7, 2011, Pages: 1581 - 1590.

Implementation of a time-warping AER mapper. Linares-Barranco, A.; Gomez-Rodriguez, F.; Jimenez, G.; Delbruck, T.; Berner, R.; Liu, S.C.; IEEE International Symposium on Circuits and Systems, 2009, Pages: 2886 - 2889.

A pencil balancing robot using a pair of AER dynamic vision sensors. Conradt, J.; Cook, M.; Berner, R.; Lichtsteiner, P.; Douglas, R.J.; Delbruck, T.; IEEE International Symposium on Circuits and Systems, 2009, Pages: 781 - 784.

An embedded AER dynamic vision sensor for low-latency pole balancing. Conradt, J.; Berner, R.; Cook, M.; Delbruck, T.; IEEE 12th International Conference on Computer Vision Workshops, 2009, Pages: 780 - 785.

Spike-based control monitoring and analysis with Address Event Representation. Jimenez-Fernandez, A.; Linares-Barranco, A.; Paz-Vicente, R.; Jimenez-Moreno, G.; Berner, R.; IEEE/ACS International Conference on Computer Systems and Applications, 2009, Pages: 900 - 906.

CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking. Serrano-Gotarredona, R.; Oster, M.; Lichtsteiner, P.; Linares-Barranco, A.; Paz-Vicente, R.; Gomez-Rodriguez, F.; Camunas-Mesa, L.; Berner, R.; Rivas-Perez, M.; Delbruck, T.; Shih-Chii Liu; Douglas, R.; Hafliger, P.; Jimenez-Moreno, G.; Ballcels, A.C.; Serrano-Gotarredona, T.; Acosta-Jimenez, A.J.; Linares-Barranco, B.; IEEE Transactions on Neural Networks, Volume 20, Issue 9, 2009 , Pages: 1417 - 1438.

32-bit Configurable bias current generator with sub-off-current capability. Delbruck, T.; Berner, R.; Lichtsteiner, P.; Dualibe, C.; IEEE International Symposium on Circuits and Systems, 2010, Pages: 1647 - 1650.

Temporal contrast AER pixel with 0.3%-contrast event threshold. Delbruck, T.; Berner, R.; IEEE International Symposium on Circuits and Systems, 2010, Pages: 2442 - 2445.

Fully integrated 500uW speech detection wake-up circuit. Delbruck, T.; Koch, T.; Berner, R.; Hermansky, H.; IEEE International Symposium on Circuits and Systems, 2010, Pages: 2015 - 2018.