ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

University of Alberta

DESIGN OF COMPUTATIONAL CMOS IMAGE SENSORS

by

David Chung-Yu Li ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2005

To my parents;
my brother, Raymond;
and Phoebe.

Abstract

CMOS image sensors have been gaining popularity in digital cameras. CMOS image sensors are also used in areas other than digital photography because it is possible to integrate many functions onto the same chip. This thesis develops circuit methods allowing the development of computational CMOS image sensors that include on chip signal processing. A 0.35 μm CMOS image sensor chip containing 8 different pixels is fabricated and the test results and analysis are presented in this thesis. The thesis also proposes a computational CMOS image sensor design that uses pulse width modulation (PWM) to process information. The proposed architecture uses simple logic gates to perform signal processing tasks, such as edge detection. By placing the signal processing module close to a pixel, the amount of information to be read out from the imager can be significantly reduced.

Acknowledgements

I am grateful for the financial support for this research from the Faculty of Engineering and the Natural Sciences and Engineering Research Council of Canada (NSERC).

I would like to thank my supervisors, Dr. Vincent Gaudet and Dr. Anup Basu, for their advice, guidance and support throughout this research. I would also like to thank my colleagues for their assistance and discussions: Dave Nguyen, Edmund Fung, Siavash Sheikh Zeinoddin, Lei Zhu, Tyler Brandon, Dan Leder and Chris Winstead.

Finally, I would like to thank my family for their support, and my girlfriend Phoebe for her love and encouragement.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ADC	Analog to digital convertor
AGC	Automatic gain control
APS	Active pixel sensor
ASIC	Application specific IC
ASP	Analog signal processor
CCD	Charge coupled device
CDS	Correlated double sampling
CFA	Color filter arrays
CMC	Canadian Microelectronics Corporation
CMOS	Complementary metal oxide semiconductor
DPS	Digital pixel sensor
FPGA	Field-programmable gate array
FPN	Fixed pattern noise
IC	Integrated circuit
JPL	Jet Propulsion Laboratory
MOS	Metal oxide semiconductor
MSB	Most significant bit
MUX	Multiplexer
PCB	Printed circuit board
PFM	Pulse frequency modulation
PPS	Passive pixel sensor
PWM	Pulse width modulation
QE	Quantum efficiency
RAM	Random access memory
SNR	Signal to noise ratio
TFA	Thin film on ASIC
VHDL	VHSIC hardware description language
VHSIC	Very high speed integrated circuits
VVL	VLSI Vison Ltd.

List of Symbols

E_{ph}	Energy of a photon
h	Planck's constant
c	Speed of light
λ	wavelength of light
$G(x)$	Rate of e-h pairs generated
x	Depth from the surface of semiconductor
α	Absorption coefficient
I_0	Incident photon flux at the surface (x=0)
J_{tot}	Total current density
J_{drift}	Drift current density
J_{diff}	Diffusion current density
q	Charge
L_n	Diffusion length of excess electrons in the p region
D_n	Diffusion coefficient for electrons
W	Width of depletion region
n_{p0}	Equilibrium minority carrier concentration

Chapter 1

Introduction

1.1 Motivation

Today, image sensors are increasingly used in consumer products. Digital cameras, web cameras, optical mouses, and smart sensors are some examples. Digital cameras are becoming very common and very popular among the average consumers. The reason for this choice is that digital cameras now have good resolution and dynamic range comparable to conventional 35mm film cameras and the images in digital cameras are easily manageable. Electronic image capturing devices have been developed since the 1960s [31], with charge-coupled devices (CCD) being the prevailing technology from the 1970s [3] until now. CCDs have the advantages of good sensor performance which includes low noise, high quantum efficiency, and low dark current. By the late 1990s, complementary metal oxide semiconductor (CMOS) image sensors started to replace CCDs in many applications requiring low power. CMOS image sensors have many advantages such as low voltage operation, low power consumption and on-chip functionality such as analog signal processing (ASP) and analog-to-digital converters (ADC), and lower cost, as compared to CCD.

One of the most used CMOS image sensor architectures is based on the active pixel sensor (APS). The APS circuit was developed in the early 1990s by the U.S. Jet Propulsion Laboratory [9] and since then many researchers have made modifications to the APS circuit to improve its performance. As the number of pixels and dynamic range of an image sensor increase to provide better image quality, the amount of information that needs to be processed also increases. This contributes to the increase of the bandwidths needed to transfer the image information in constant time to an image processing unit, such as a computer. The increase of information will also increase the processing time.

This thesis develops a new pixel architecture that could improve performance by putting a signal processing unit inside or near each pixel by using a different type of computation in the signal processor units. Newer sensor technologies, such as thin film on ASIC (TFA) technology which allows the photodiode to be placed on top of metal layers [22], are allowing computation modules to be placed at pixel-level. Design and test results of a CMOS image sensor integrated circuit (IC) are presented. A new architecture for a CMOS imager based on pulse width modulation (PWM) is designed and simulated in this work.

1.2 Thesis Outline

This thesis contains 6 chapters. Chapter 2 gives background information on solid-state imagers, including a survey of existing designs of CMOS imagers. Chapter 3 describes the design and testing of a prototype imager chip. Chapter 4 gives background information on statistical theories and on signal processing. Chapter 5 describes the design of a computational CMOS imager. Chapter 6 concludes this thesis and presents suggestions for future work.

Chapter 2

Background on solid-state image sensors

This chapter describes the basics of CMOS image sensors. Section 2.1 presents a brief historical background and a comparison of CCDs and CMOS image sensors. Section 2.2 describes the basic operation and the circuitry used in CMOS image sensors. Section 2.3 presents a survey of state-of-the-art CMOS image sensor designs and their applications.

2.1 Solid-state image sensor background

2.1.1 Historical background

Fossum discusses the historical development of MOS (metal oxide semiconductor), CCD, and CMOS image sensors in [9]. This is summarized in the following section.

Timeline of MOS, CCD and CMOS image sensors development

1963 - Morrison at Honeywell designed a light sensor that is similar to a computational sensor [31].

1964 - Horton *et al.* at IBM reported a solid-state image scanner called the scanistor [16].

1967 - Weckler proposed using photodiodes as charge integrating photodetectors [48]. A reverse-biased photodiode is used and the photocurrent is integrated on the photodiode capacitance. Weckler also suggested using a PMOS switch for readout.

1968 - Noble described several types of image sensors, surface and buried photodiodes, and a charge amplifier for readout [34]. Also, he showed the use of a MOS source-follower transistor in the pixel for readout buffering.

1970 - Fry, Noble, and Rycroft reported fixed-pattern noise (FPN) in image sensors [10].

1970 - CCD was invented by Boyle and Smith at Bell Labs [3]. Its low FPN was one of the many advantages over other types of solid-state image sensors.

1970's to 1980's - Researchers put a lot of effort into improving CCD performance during this period. There are huge improvements in areas such as quantum efficiency, dark current, fill factor, readout rate, readout noise, and dynamic range. MOS image sensors were hardly researched and their performance was not good compared to CCD [25].

Late 1970's to early 1980's - Hitachi and Matsushita continued to develop MOS image sensors [36], [41]. In 1979, Hitachi reported the first commercial single-chip color MOS image sensor [19].

Early 1990's - Researchers at the University of Edinburgh in Scotland (later becoming VLSI Vision Ltd. or VVL) started a resurgence to create highly functional single-chip low cost imaging systems. U.S. Jet Propulsion Laboratory (JPL) started research on imaging systems that are driven by performance and can be used in space application. This effort led to the development of the CMOS active pixel sensor (APS).

1990 - CMOS passive pixel sensors (PPS) were developed by VVL [40].

1994 - JPL reported the CMOS APS [26].

Mid 1990's to now - CMOS technology continues to improve. The main advantages of CMOS image sensors over CCD systems are on-chip functionalities, lower system power, and lower system cost. Today, APS is the most common pixel architecture used by many CMOS image sensor designs.

2.1.2 Charge Coupled Devices and CMOS image sensors

CCD and CMOS cameras are electronic systems that can convert light to digital information to capture an image. In a CMOS image sensor, the process of converting light to digital information consists of five steps. First, light waves are collected into pixels in the image sensor through microlenses. These light waves are then separated into red, green, and blue light waves by passing through color filters. The light waves are then converted to an analog voltage or current level by a photodetector and other circuits in each pixel. After the conversion, this analog voltage in each pixel is transferred out of the array and further processed by an analog signal processing unit. Lastly, the processed analog voltage is converted to digital information by an ADC. The digital information can then be transferred to a computer for further processing and storage.

A CCD is a dynamic analog shift register implemented using closely spaced MOS capacitors clocked using 2, 3, or 4 phase clocks [12]. The block diagram of a CCD camera system is presented in Figure 2.1. A majority of today's CCD image sensors use interline transfer. In interline transfer CCD, photodiodes are used as photodetectors. All CCDs are optically shielded and used only for readout. The collected charge from the photodiode is simultaneously transferred to the vertical CCDs at the end of integration time and then shifted out. The vertical CCDs are shifted out one row at a time to the horizontal CCDs and these horizontal CCDs are shifted out one at a time to the amplifier. The value of each CCD is passed through a correlated double sampling (CDS) unit to reduce noise, an automatic gain control (AGC) unit, an ADC, and a signal processor for color correction and other processing functions.

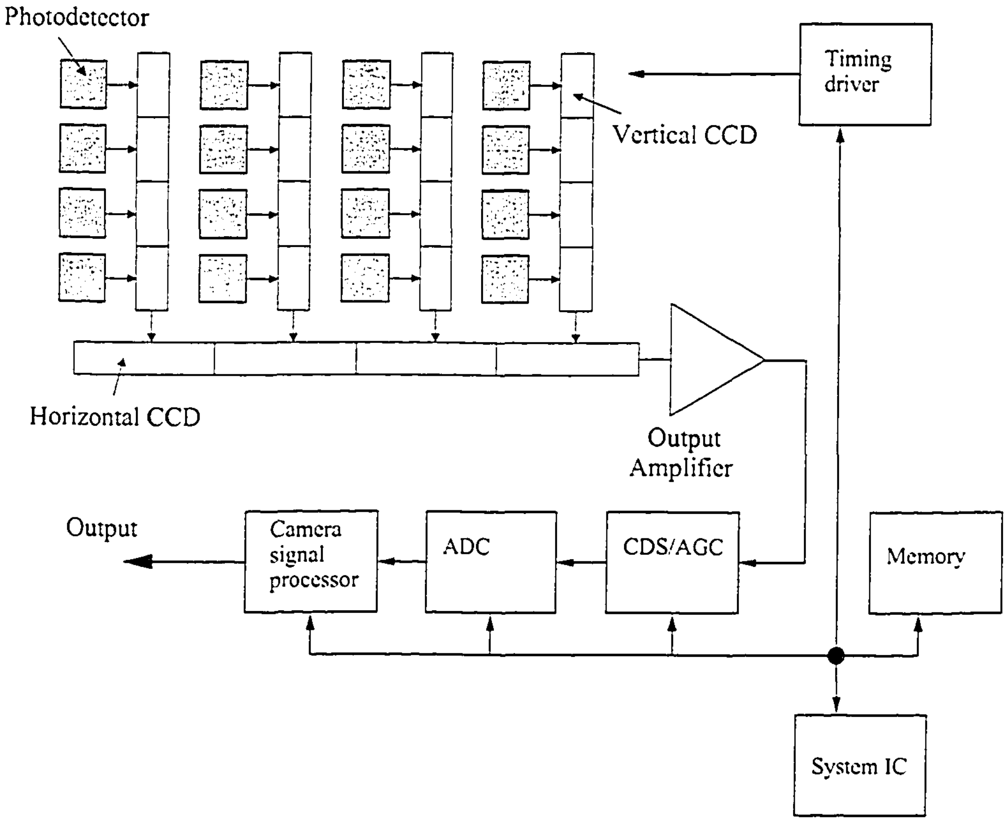


Figure 2.1: Block diagram of a CCD camera (Interline Transfer)

A major difference between CMOS image sensors and CCDs is the readout architecture. In CMOS image sensors, charge or voltage (or current) is read out using column and row circuits and in CCDs, charge is shifted out. A block diagram of CMOS image sensors is presented in Figure 2.2 [12]. The typical readout mode is to transfer one row at a time to the ASP for functions such as CDS and gain. Each row is read out by reading one or more pixels at a time using a column selector and multiplexer. Other modes of image readout are also available. A window readout mode and a skip readout mode can be used. In window readout mode, a small region of pixels is selected for readout. This increases access rates to windows of interest. In skip readout mode, every 2nd, or 3rd pixel is read out. This mode can increase readout speed by reducing the resolution. Electronic pan, tilt and zoom can be implemented by using a combination of skip and window modes.

Another major difference is that CCDs cannot be produced using standard CMOS processes and therefore, many different external components such as ADC, memory and timing drivers are required for timing and signal processing. The shift-style readout of the CCD uses more power because the entire array of pixels is always switching. The main advantage of CCDs is that they have high quantum efficiency (QE) and low dark current. QE is the ratio of electrons (or holes) collected to incident photons. On the other hand, CMOS image sensors can incorporate many functionalities on one chip thus reducing component and packaging costs. The main disadvantage of CMOS image sensors is higher noise such as FPN. Advantages and disadvantages are summarized in Table 2.1. For applications that require high image quality, CCDs will be the choice to use. If the emphasis is on functionality, high speed, low cost and low power, then CMOS image sensors will be the better choice. In this thesis, the focus is on CMOS image sensors because the technology allows us to add functionality at the pixel level.

2.2 Description of CMOS image sensors

2.2.1 Microlens and color filters

A Microlens is a layer of glass that focuses light to the photosensitive portion of each pixel. The microlens is implemented by using a polymer-thin film or by etching the glass deposited on the chip. This process is not a part of the fabrication process of a CMOS image sensor chip and needs to be done separately. Microlenses can improve

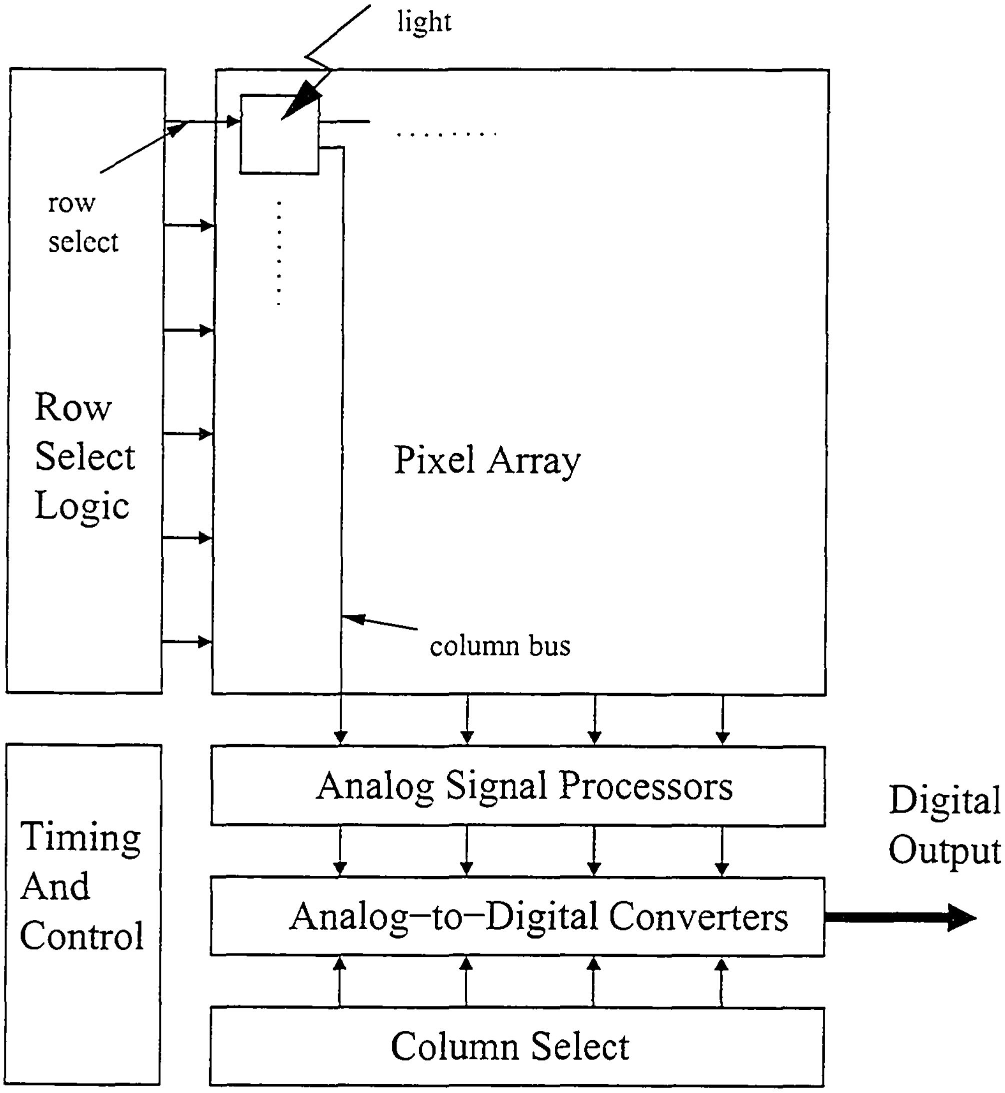


Figure 2.2: Block diagram of a CMOS image sensor

Table 2.1: Summary of advantages and disadvantages of CCD and CMOS image sensors [12]

Type of image sensor	Advantages	Disadvantages
CMOS	- can integrate other camera functions on same chip - lower power consumption than CCDs (10x) - very high frame rates can be achieved	- nonoptimized photodetectors - high noise due to multiple levels of amplification (pixel, column, and chip) - FPN
CCD	- optimized photodetectors - high QE - low dark current - very low noise, very low FPN because no noise and no FPN are introduced during shifting	- difficult to integrate other camera functions on same chip - difficult to implement window of interest - high power because entire array is switching all the time - limited frame rate due to serial readout

the fill factor, which is the ratio of the sensitive region to the total pixel area, by about 2 to 3 times compared to image sensors without microlens. After light passes through the microlens, it passes through a layer of color filter arrays (CFA) which separate the colors of the image. Figure 2.3 shows the cross section view of a pixel in the CMOS image sensor.

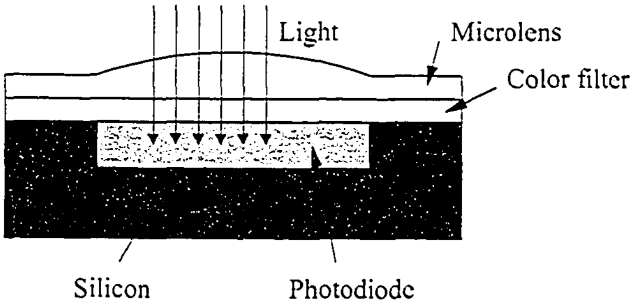
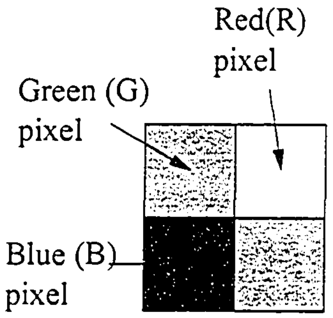


Figure 2.3: Cross section of a pixel in CMOS image sensor

The image sensor must be able to separately measure red (R), green (G), and blue (B) light in order to capture a color image. To do this, each pixel of a color image sensor is covered with a red, green, or blue filter, according to a specific pattern. The Bayer CFA pattern is one of the most common patterns used and it has a repeating 2x2 arrangement [38]. Figure 2.4 illustrates the pattern for the Bayer CFA. CFA performance such as spectral response and noise was described in [7].



G R G R G R G R
B G B G B G B G
G R G R G R G R
B G B G B G B G

Figure 2.4: Bayer CFA pattern

In Figure 2.4, when the image is read out, line by line, the pixel sequence comes out GRGRGR and so on, and then the alternate line sequence is BGBGBG and so on. Since each pixel can only detect one color, color interpolation is needed to determine the amount of light for the other two missing colors to form the complete color image. Color interpolation is usually done on chip using digital signal processing techniques [38].

2.2.2 CMOS image sensor system-level architecture

A typical single-chip CMOS image sensor is usually composed of 6 blocks, as illustrated in Figure 2.2. An array of pixels, with width and height between 5 – 10 μm, are used to convert light into electrical signals. The CMOS image sensor reads out the signal collected by each pixel by directly addressing it, in a way that is analogous to the way a random access memory (RAM) works. There are the row decoder and the column select lines, which are used to select a row and column for readout. The pixel array contains all the light sensing pixels and the output of each column of pixels will then go to the analog signal processor (ASP) block. The ASP is used to provide noise reduction and removal of artifacts such as fixed-pattern noise. The output ADC is used to provide a digital interface for the image sensor. More signal processing is done in the digital domain outside the chip, such as color correction. The most common way to get the output of the pixels is to read out the information in a row by row fashion throughout the entire array.

2.2.3 Design and operation of an active pixel sensor

One of the most common pixel structures for CMOS image sensors is the active pixel sensor (APS) pixel. The APS circuit was developed in the early 1990s by the U.S. Jet Propulsion Laboratory [9] and since then many researchers have made modifications to the APS circuit to improve its performance. The standard APS pixel contains 3 transistors and a pn junction photodiode, as shown in Figure 2.5. Reset transistor (M1) is used to control the charge-sensing node at N1. Amplifier transistor (M2) is connected as a source follower with an off-pixel current source (M4). Row select transistor (M3) is used to select the pixel and to buffer the pixel output to the column bus for readout.

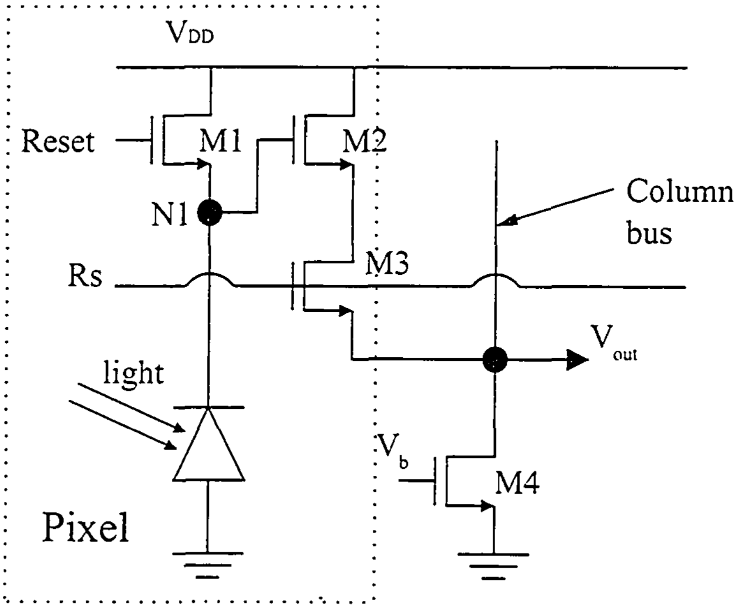


Figure 2.5: APS structure

There are 3 stages in the operation of the APS: reset, integration, and readout. Figure 2.6 is a SPICE simulation using the SpectreS simulator that shows the 3 stages of the APS operation. The currents in the figure were selected from typical values listed in [29]. In the reset stage, reset and row select transistors are on and node N1 is pulled to a voltage of V_{dd}-V_{tn}, where V_{dd} is the supply voltage and V_{tn} is the threshold voltage of the transistor. The photodiode is reverse biased at V_{dd}-V_{tn}. In the integration stage, reset and row select transistors M1 and M3 are turned off. When light shines on the photodiode, the photodiode produces a reverse current that is directly proportional to the light intensity. Equations 2.5 - 2.8 show how this photocurrent is calculated. The current density multiplied by the area of the photodiode gives the photocurrent. The photodiode discharges the capacitance of N1 for t_{int} seconds, which is called the integration time. Section 2.2.6 shows how the node voltage N1 is calculated. The current i is composed of photocurrent and dark current. Dark current is the photodiode leakage current and it is not induced by photogeneration. The capacitance of N1 (C_{N1}) is often composed of the photodiode's own capacitance plus any connected device capacitances.

$$Q = i * t_{int} \tag{2.1}$$

$$V_{N1} = V_{dd} - V_{tn} - \frac{Q}{C_{N1}} \tag{2.2}$$

For the readout stage, the row select transistor is turned on. Voltage at node N1 is sampled and transferred to the column bus for analog signal processing. The readout period is determined by the lighting condition of the environment. If the environment is bright, which is when V_{out} is low, then the readout will be earlier so that the pixel will not be saturated or V_{out} = 0V. In the opposite condition, if it is dark, which is when V_{out} is high, then the readout will be later so that the integration period is longer for improved sensitivity.

More recent CMOS image sensor designs use fewer than 3 transistors per pixel. In [30], a design for an image sensor with 7 transistors per four pixels or 1.75 transistors/pixel is presented. This design has a group of 4 photodiodes sharing a floating diffusion (FD) region, a reset transistor, an amplifying transistor and a select transistor. The pixel size is very small with a size of 2.25 x 2.25 μm^2 and it is manufactured in 0.25 μm CMOS process. Similar to [30], Takahashi *et al.* [44] designed an image sensor with 1.5 transistors/pixel by using one transistor for both reset and row select. The pixel size is 3.9 μ x 3.9 μ and it is fabricated in 0.35 μm CMOS process.

2.2.4 Analog signal processors and analog-to-digital converters

After the node N1 voltage is sampled onto the column bus, the analog signal is then passed to the ASP. The ASP units perform functions such as gain, sample and hold, CDS and FPN suppression. When the ASP unit finished processing the signal, this signal is converted to a digital value using an ADC. The precision of

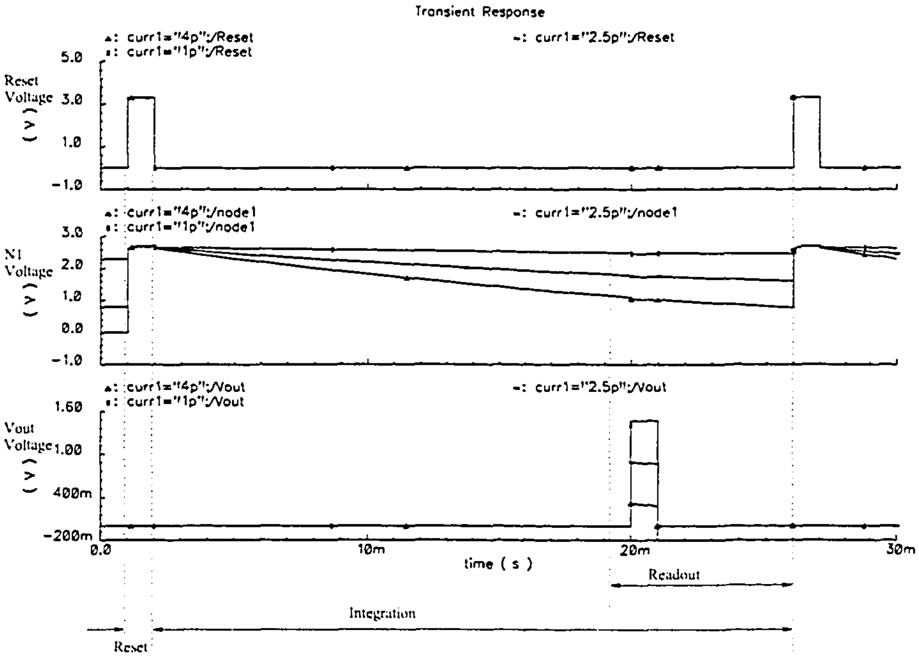


Figure 2.6: Timing diagram of the APS operation

the value depends on the dynamic range of the ADC. For example, if the precision of the ADC is 8 bits, then the output value for each pixel consists of 8 bits and is limited by noise. The digital output is transferred to a digital signal processor or a computer for more computation. Normally, the ASP and ADC units take up about one-third to half the area of a CMOS camera chip. With the increase of scale of image sensors, the bandwidth needed to output the information in fixed time to the ASP and ADC units also increases. A possible way to reduce the bandwidth requirements is to complete some of the signal processing function inside or near the pixel, so that the amount of information can be reduced for output.

The objective of the thesis is to develop an image sensor with the capability of simple calculations at the pixel in order to improve bandwidth.

2.2.5 Photodiode design and operation

There are many types of photo sensors available, such as photodiodes, photogates, bipolar transistors [45] and charge modulation devices [35]. The bipolar and charge modulation device image sensors require specialized fabrication processes, whereas photodiodes and photogates can be fabricated using standard CMOS processes. Photodiodes and photogates have lower noise, better uniformity and simpler layout compared to bipolar transistors and charge modulation devices. Photogates are used in frame transfer CCDs and CMOS Photogate APS, but they have lower spectral response, especially for blue wavelengths, due to the absorption in the polysilicon gate. Therefore, the photodiode has been widely used in CMOS image sensor designs [18, 32, 50, 53]. Standard CMOS N-well processes can support three pn junction photodiode types: N+ diffusion to P-type substrate (n^+ - p_{sub}), N-well to P-type substrate (n_{well} - p_{sub}), and P+ diffustion to N-well (p^+ - n_{well}) [39]. Figure 2.7 shows the cross section view of the 3 different types of photodiodes. The n_{well} - p_{sub} photodiode is more sensitive to longer wavelengths such as red light because of the depth of the n_{well}. The n^+ - p_{sub} photodiode is more sensitive to shorter wavelengths such as blue light [54]. An important factor in designing the pixel and photodiode is the fill factor. Fill factor is the percentage of pixel area that is light sensitive. A larger photodiode area increases the diode capacitance and the full well capacity.

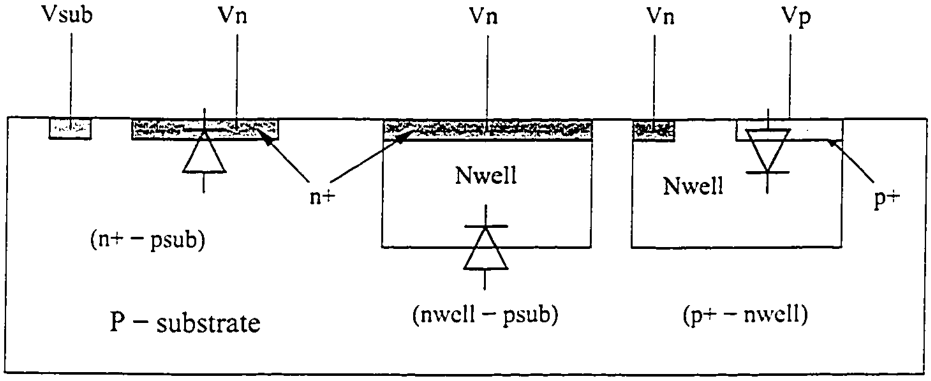


Figure 2.7: Cross section of different types of CMOS pn junction photodiodes

2.2.6 Photogeneration in silicon

In semiconductors, when an incident photon has an energy greater than the bandgap of the material, it will excite an electron to jump from the valence band to the conduction band, thus leaving a hole behind. For silicon, photons in the visible range have enough energy to generate electron-hole (e-h) pairs.

The energy of photons can be calculated by Equation 2.3, where $h = 4.135 * 10^{-15}$ eV is Planck's constant, $c = 3 * 10^8$ m/s is the speed of light, and λ is the wavelength in μm.

$$E_{ph} = \frac{hc}{\lambda} = \frac{1.24}{\lambda(\mu m)}(eV) \tag{2.3}$$

For a fixed wavelength, the rate of e-h pairs generated at a depth of x from the surface is given by Equation 2.4 [43], where I_0 is the incident photon flux at the surface (x = 0) with units of $\frac{photons}{cm^2 * s}$, and α is the absorption coefficient in cm^{-1} and is a function of λ.

$$G(x) = \alpha I_0 e^{-\alpha x} \tag{2.4}$$

A photodetector is used to convert the generated e-h pairs, into photocurrent. For a photodiode, the photocurrent consists of two parts, the drift current in the depletion region and the diffusion current out of the depletion region [43].

$$J_{tot} = J_{drift} + J_{diff} \tag{2.5}$$

Figure 2.8 shows a cross section of a photodiode and W is the width of the depletion region.

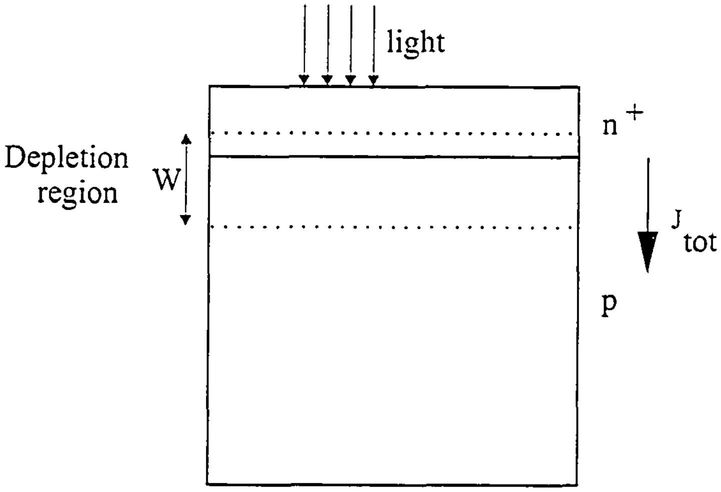


Figure 2.8: Cross section of a photodiode

Assuming that the n^+ region of the photodiode is thin enough to cause negligible absorption, and dark current is ignored, by using Equation 2.4, the drift current density is:

$$J_{drift} = q\int_0^W G(x)dx = qI_0(1 - e^{-\alpha W}) \tag{2.6}$$

The diffusion current is dominated by the current in the p region and the diffusion current density is:

$$J_{diff} = qI_0 \frac{\alpha L_n}{1 + \alpha L_n} e^{-\alpha W} + q n_{p0} \frac{D_n}{L_n} \qquad (2.7)$$

where L_n is the diffusion length of excess electrons in the p region, D_n is the diffusion coefficient for electrons, and n_{p0} is the equilibrium minority carrier concentration. Combining Equations 2.6 and 2.7, the total current density is:

$$J_{tot} = qI_0(1 - \frac{e^{-\alpha W}}{1 + \alpha L_n}) + q n_{p0} \frac{D_n}{L_n} \qquad (2.8)$$

The above equation shows that the current density is proportional to the incident photon flux, I_0. Although circuit designers do not have control over the parameters in the equation, one can use two junctions at different depths to obtain two different spectral responses [8]. Photocurrent is integrated over exposure time into charge and this charge is converted to voltage by a capacitance.

2.2.7 Noise sources in CMOS image sensors

The sensitivity and accuracy of CMOS image sensors is limited by non-idealities such as temporal noise, FPN, dark current and nonlinearities. Temporal noise is caused by photodetector and transistor thermal, shot, and 1/f noise. Temporal noise analysis in APS blocks has been investigated by Tian *et al.* [46]. FPN is the variation in pixel output due to device mismatches between pixels. The main type of FPN in CMOS image sensors is variations in the transistor threshold voltage. Underlying causes of FPN are variations in photodetector geometry, doping concentrations and contamination during fabrication. CDS is used in image sensors with photogate or pinned photodiode with transfer gate to reduce FPN. First, reset the pixel and store the reset signal onto a capacitor. Then, transfer the signal charge and store the signal charge onto another capacitor. Subtract the two stored values to get the output. This sampling method is correlated because the noise component of the two signals is correlated and the noise component can be subtracted out. For APS with photodiode, there is no separate output, so the signal must be read out first. This signal contains the original reset voltage on the photodiode and the noise components are different and not correlated. Therefore, this type of sampling is called double sampling and it cannot remove reset noise. Figure 2.9 shows how double sampling is implemented in APS. First, the signal voltage is sampled onto a capacitor C_S.

then the pixel is reset and the reset voltage is sampled onto the capacitor C_R. By subtracting the two voltages V_{signal} - V_{reset}, FPN from mismatches are removed.

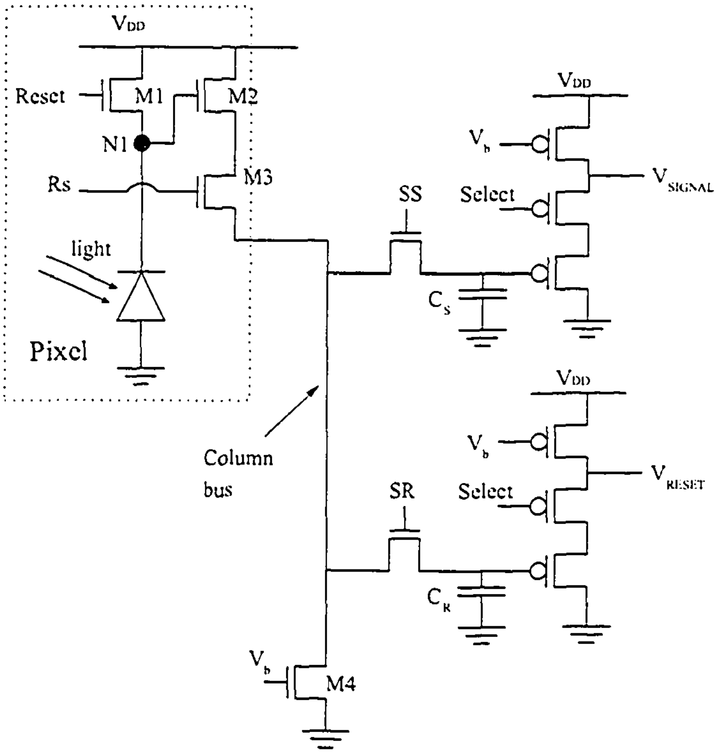


Figure 2.9: Double sampling circuit in APS

2.2.8 Signal to noise ratio and dynamic range

Signal to noise ratio (SNR) and dynamic range are very important for image sensors and are an index of image quality. SNR is the ratio of the input signal power to the average input referred noise power measured in dBs. Dynamic range provides an index of a sensor's ability to display both high and low light objects in a scene. It is defined as the ratio of the largest nonsaturating input signal to the smallest detectable input signal. Table 2.2 lists the dynamic range of several image capture devices.

2.3 Survey of CMOS image sensor designs

2.3.1 Image sensor designs and novelty

Table 2.3 summarizes various CMOS image sensor designs and Table 2.4 presents the specifications of the various designs.

Table 2.2: Dynamic range of image capture devices [12]

Type of devices	Dynamic range
Natural scenes	> 100dB
human eye	around 90dB
film	80dB
high end CCDs	> 78dB
consumer grade CCD	66dB
consumer grade CMOS sensors	54dB

Table 2.3: Different designs of CMOS image sensor

Designer	Topic	Novelty
Xu *et al.*	Complementary APS [50]	Improves the output voltage swing of the circuit in low operating voltage conditions
Kleinfelder *et al.*	Digital pixel sensor (DPS) [18]	With bit-parallel ADC and memory per pixel, DPS is fast and can reach 10000 frames/s operation
Yoon *et al.*	Image sensor for mobile application [53]	Majority of color signal processing done in analog domain, low power, uses <20mW from a 3.3V supply voltage operating at 30 frames/s
Bermak	Image sensor with pixel level PFM/PWM based ADC [2]	With on-pixel ADC based on both Pulse Frequency Modulation (PFM) and Pulse Width Modulation (PWM), features a programmable dynamic range
Decker *et al.*	Image sensor with wide dynamic range pixels [5]	Has a dynamic range of 96 dB and uses a dynamic range expansion technique providing flexible digital control of the compression characteristic
Ma and Chen	APS camera with direct frame difference output [23]	APS cell circuit has in-pixel storage for previous frame image data so that the current and the previous frame can be read out simultaneously

Table 2.4: Specifications and performance of CMOS image sensor designs

Performance	Xu *et al.*	Kleinfelder *et al.*	Yoon *et al.*
technology (μm)	0.25	0.18	0.35
pixel size (μm)	12 x 10	9.4 x 9.4	5.6 x 5.6
number of pixels	128 x 128	352 x 288	352 x 288
power supply	1V	1.8V	3.3V
power consumption	75mW (parallel) 18mW (serial)	50mW	20mW
fill factor	30%	15%	28%
frame rate	48fps (parallel) 15fps (serial)	> 10000fps	30fps
Performance	Bermak	Decker *et al.*	Ma and Chen
technology (μm)	0.25	0.8	0.5
pixel size (μm)	45 x 45	-	32.2 x 32.2
number of pixels	32 x 32	256 x 256	128 x 96
power supply	-	5V	3.3V
power consumption	85μW (PFM) 10μW (PWM)	52mW @ 30fps 400mW @ 390fps	56mW
fill factor	23%	49%	33%
frame rate	-	390fps	250fps

Researchers are trying to develop image sensors that have many on chip functions [14], [28], [53]. Others, as in [23], [32], [37], use analog in-pixel processing to reduce the size of the chip and the power consumption. These small and low power image sensors can be used in many applications such as cell phones and cars. Researchers are also trying to incorporate the ADC in the pixel to increase speed [18], [49], [52]. Image sensors with wide dynamic range are found in [5], [42], [51]. The logarithmic APS are developed for the purpose of increasing the dynamic range. A logarithmic APS pixel is show in Figure 2.10. The difference between this pixel and the standard APS pixel is that the gate of the reset transistor is tied to V_{DD}. The advantages of Logarithmic APS is that the sensor measures illumination over a range of more than 5 orders of magnitude [6], and a dynamic range of over 100dB. Also, there is no need for a reset line and no integration time, which means the pixel can be read out at any time. The disadvantages are high FPN and the small output signal swing makes the sensors susceptible to other noise sources.

New ways to detect color are also being developed. In [8], Findlater *et al.* designed an image sensor that employs a vertically integrated double-junction photodiode structure and only uses a 2 color filter, instead of the traditional CFA that

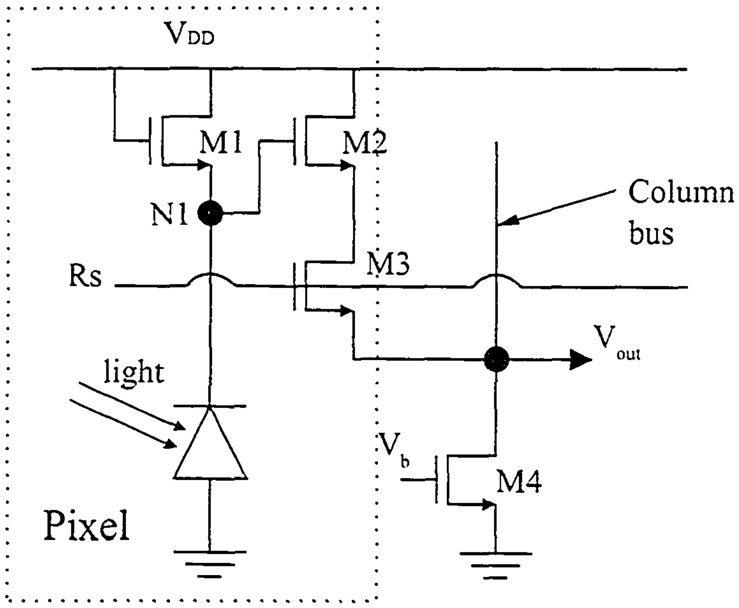


Figure 2.10: Logarithmic APS structure

contain 3 colors. Foveon [27] designed a chip where each pixel can detect all colors. Each pixel can detect red, green, and blue, depending on how far the photon penetrates through different layers of silicon. Each pixel has 3 pn junctions by stacking 3 n_{well}s. This type of design eliminates the need for CFA and the calculations necessary to find the missing information in an image, but requires non-standard processes.

2.3.2 Computational image sensors

Computational sensors [17] are the product of integrating sensing and processing on an IC chip. This type of sensor has the potential to reduce latency through parallel computation. Previous work with computational CMOS image sensors are discussed here.

On-sensor compression

Aizawa *et al.* reported a computational image sensor with on-sensor compression [1]. This compression sensor uses a conditional replenishment algorithm. The current pixel value is compared to the last value stored in the memory. If the difference is greater than a specified threshold, then the current value and address of the pixel are outputted. The value of the memory of the activated pixel is replenished by the current input value, and if the pixel is not active, then its value is kept unchanged. The compression sensor contains the basic APS circuit, plus the computational elements of a memory and a comparator. With this architecture, the fill factor is much smaller than the standard image sensors because the computational elements take up some area.

Adaptive pixel-based integration time

Hanamoto and Aizawa presented a computational image sensor with adaptive pixel-based integration time [15]. Each pixel in the sensor can independently control its integration time to the intensity of incident light and to the temporal changes due to motion. The integration time of each pixel is shortened if motion is detected in the pixel or pixel intensity becomes saturated. This reduces image blur caused by moving objects in a scene as well as improving intrascene dynamic range. Most of the processing circuit is shared by the pixels of a column. The sensor activates flag signals when it detects motion or saturation of the pixels and outputs only the activated pixels. Due to the variable integration time, the pixel intensity needs to be normalized afterwards.

Signal processing using a simple analog operation

Muramatsu *et al.* reported a high-density image sensor with three signal processing function modes: wide dynamic range, motion detection, and edge extraction [32]. They have developed three component technologies to perform the three signal processing function modes. Firstly, an on-pixel frame memory consists of a transistor gate capacitor and a transfer transistor. Secondly, a double-sampling readout scheme, with two exposure times for one-frame readout, provides a nondegraded frame-readout rate with interframe analog operation. Finally, a column-parallel CDS circuit is used as a subtract circuit to provide column-parallel, on-chip analog operation.

Pixel level PWM signal processing

Nagata *et al.* proposed a smart imager with pixel level pulse width modulation (PWM) signal processing [33]. This imager performs row parallel single or multi-bit PWM readout, block averaging, and 2D projection of a thresholded image. The block averaging and 2D projection are realized with PWM signal addition techniques based on switched current integration and charge packet counting. For the block averaging, the PWM voltage pulses are converted to current pulses by the switched current sources. These current pulses are integrated on a capacitor to give a sum of pixel values. A charge packet counter is used to convert a small reference charge to a current pulse and removes it from the capacitor successively during the integration. An averaged result is calculated from the pulse counts. For the 2D projection, pixels are compare to a reference voltage to generate voltage pulses. The horizontal projection values are obtained as the number of pulses counted by the row counters. The vertical projection values are obtained by the switched current integration, and the use of the charge packet counter.

2.4 Chapter conclusion

We have reviewed the history and architecture of CMOS image sensors, the standard APS pixel and some variations on the design. Computational image sensors were discussed briefly. The focus of this thesis is to develop a computational image sensor where the signal processing is close to the pixel. In order to understand more design tradeoffs, a prototype chip was designed to evaluate different pixel architectures; it will be discussed in the next chapter.

No Text

Chapter 3

CMOS image sensor prototype chip

Chapter 2 described state of the art CMOS imager designs and their applications. In this chapter, the design and testing of a prototype CMOS pixel chip are discussed. We will also discuss the tests that were performed to model the various pixels in the chip and the test results.

3.1 The design of the chip

The prototype chip is composed of 8 different pixels and a 3-to-8 row decoder. This chip does not contain any signal processors or analog-to-digital converters. In Figure 3.1, there are two different APS architectures. The one on the left is the original design and consists of 3 transistors and a photodiode. The one on the right is similar to the left one, but with an extra transistor between M1 and the photodiode. By adding an extra transistor, the charge to voltage conversion gain can be increased [20]. The sensor can either operate in gain-enhancement mode if the gate voltage of M4 is at a lower value such as 1.5V or it can operate in normal mode if the gate voltage of M4 is at V_{DD} [47].

The 8 different pixels are based on the 2 APS architectures with some variations. These variations are 2 types of reset transistor M1, N_{mos} or P_{mos}, and 2 types of photodiodes, n^+ - p_{sub} and n_{well} - p_{sub}. Table 3.1 shows the 8 different pixels and their respective properties. The size of the photodiode is a square of 6μm by 6μm and the size is the same for all 8 pixels. However, the pixel sizes are not the same, they range from 12.5μm by 8μm to 12.2μm by 14.6μm. For this chip, uniform size and shape for the photodiodes are the most important controlled factors and not

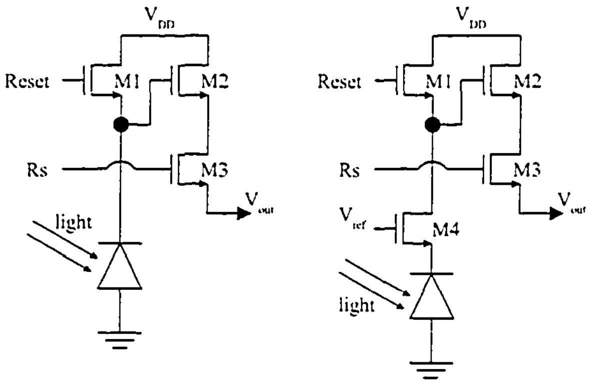


Figure 3.1: Different APS structures

the overall shape of the pixel. Figures 3.2 to 3.5 show the layout of the 8 pixels. The prototype chip was manufactured in a standard 3.3V 0.35μm 1-poly 4-metal CMOS process. A grant of fabrication area was provided by Canadian Microelectronics Corporation (CMC). A total of 5 packaged dies and 40 loose dies were received, and the package dies were tested.

Table 3.1: List of the 8 pixels and their properties

Pixel	Number of transistors per pixel	Reset transistor type	Photodiode type	Pixel size (μm)	Fill factor (%)
1	3	N_{mos}	n^+ - p_{sub}	12 x 8.5	35
2	3	N_{mos}	n_{well} - p_{sub}	12 x 8	38
3	3	P_{mos}	n^+ - p_{sub}	11.2 x 13.1	25
4	3	P_{mos}	n_{well} - p_{sub}	10.7 x 14.6	23
5	4	N_{mos}	n^+ - p_{sub}	8.4 x 13.5	32
6	4	N_{mos}	n_{well} - p_{sub}	8.4 x 13.5	32
7	4	P_{mos}	n^+ - p_{sub}	10.9 x 13.6	24
8	4	P_{mos}	n_{well} - p_{sub}	12.2 x 14.6	20

The row decoder was designed using a pass transistor network, inverters, and pull-up transistors. The transistors' lengths are 0.35 μm and widths are 0.8 μm. Figure 3.6 shows the schematic diagram of the row decoder.

Total area of the chip including bonding pads is 631μm by 503μm. Figure 3.7 shows the layout of the prototype chip without bonding pads. Figures 3.8 and 3.9 show die microphotographs of the chip.

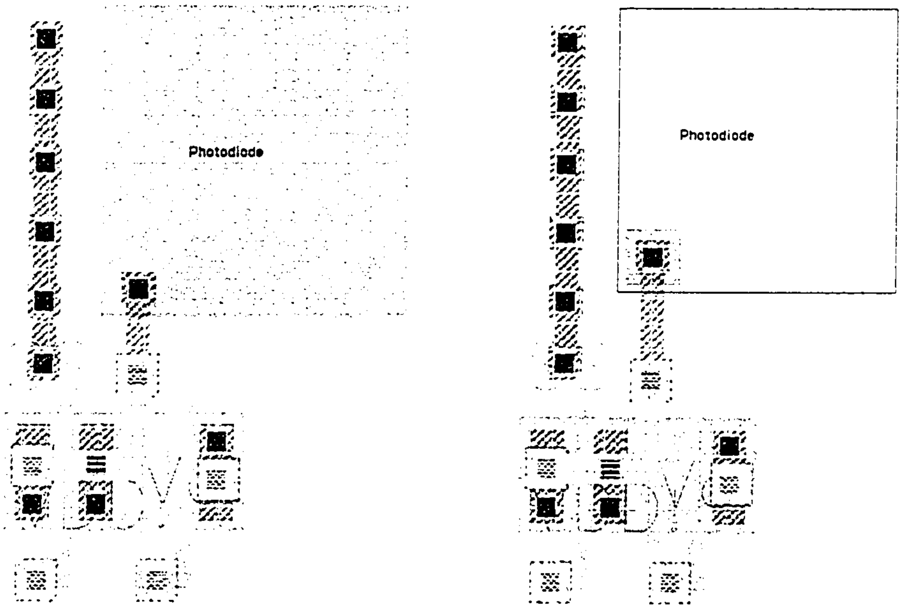


Figure 3.2: Layout of pixels with 3 transistors and N_{mos} reset, with n^+ photodiode on the left and n_{well} photodiode on the right

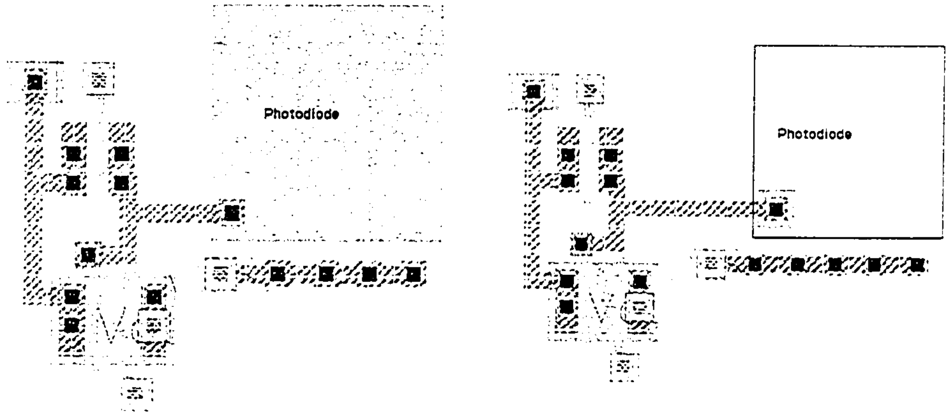


Figure 3.3: Layout of pixels with 3 transistors and P_{mos} reset, with n^+ photodiode on the left and n_{well} photodiode on the right

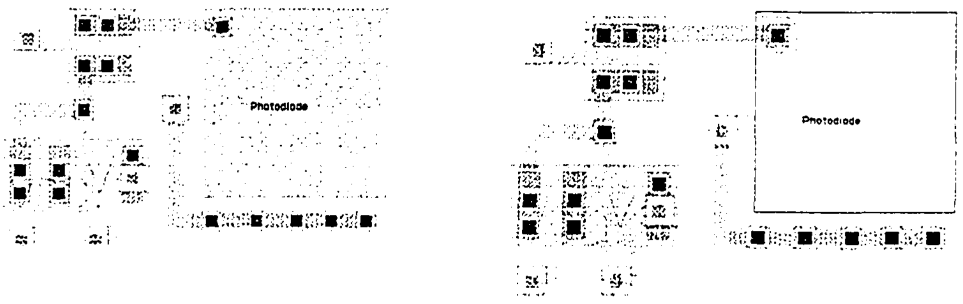


Figure 3.4: Layout of pixels with 4 transistors and N_{mos} reset, with n^+ photodiode on the left and n_{well} photodiode on the right

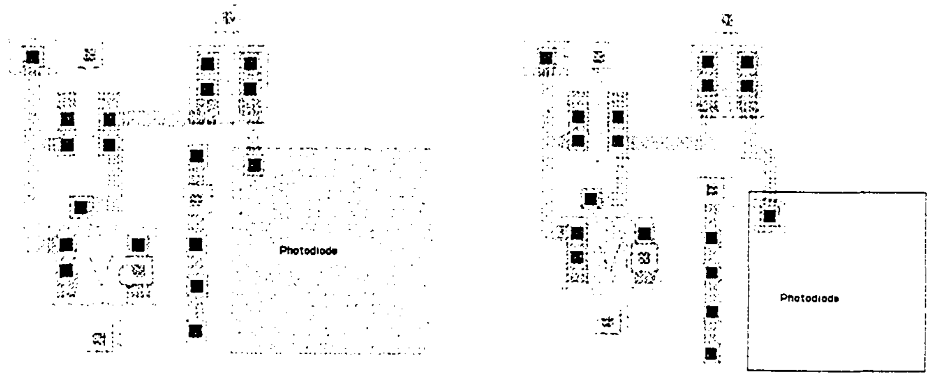


Figure 3.5: Layout of pixels with 4 transistors and P_{mos} reset, with n^+ photodiode on the left and n_{well} photodiode on the right

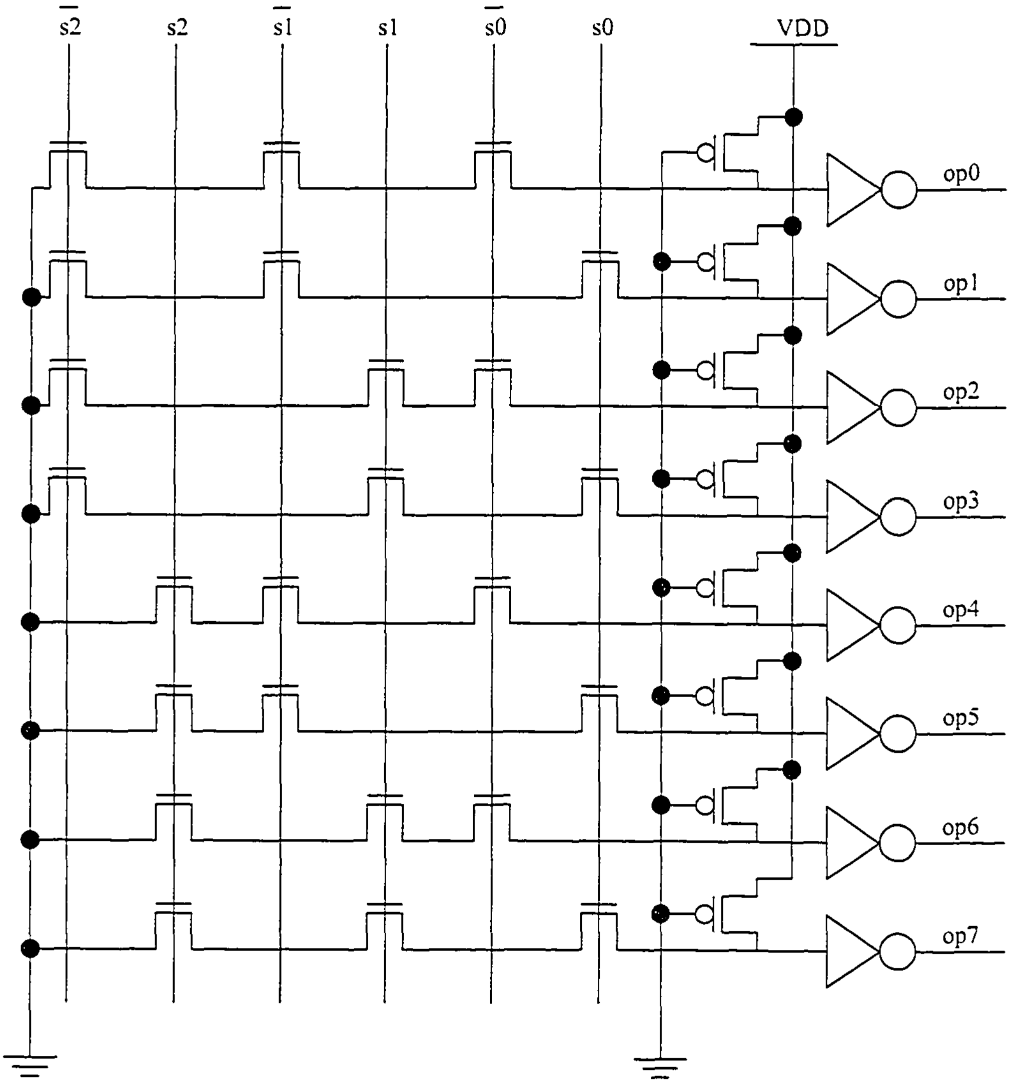


Figure 3.6: Schematic of row decoder

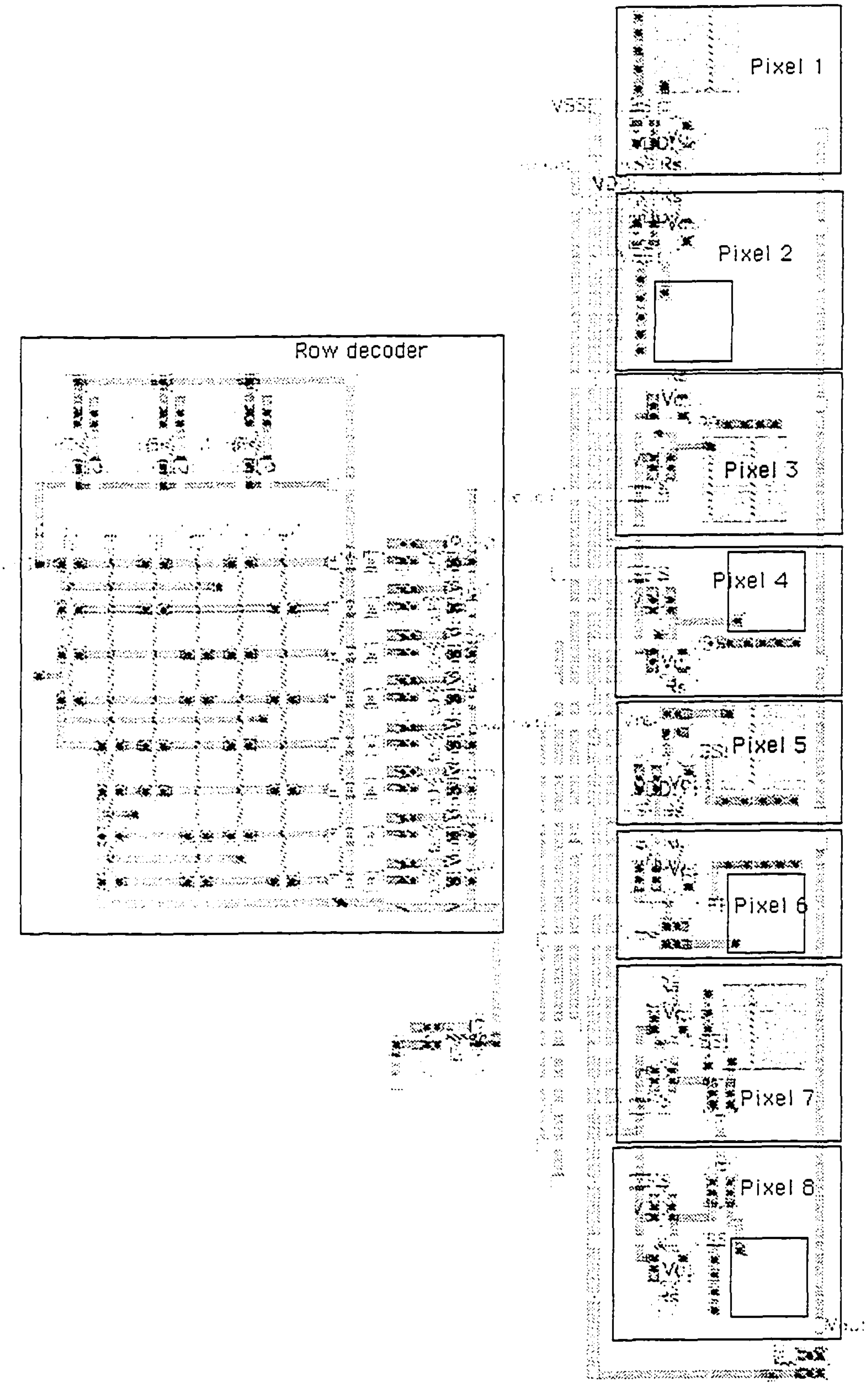


Figure 3.7: Layout of prototype chip without bonding pads

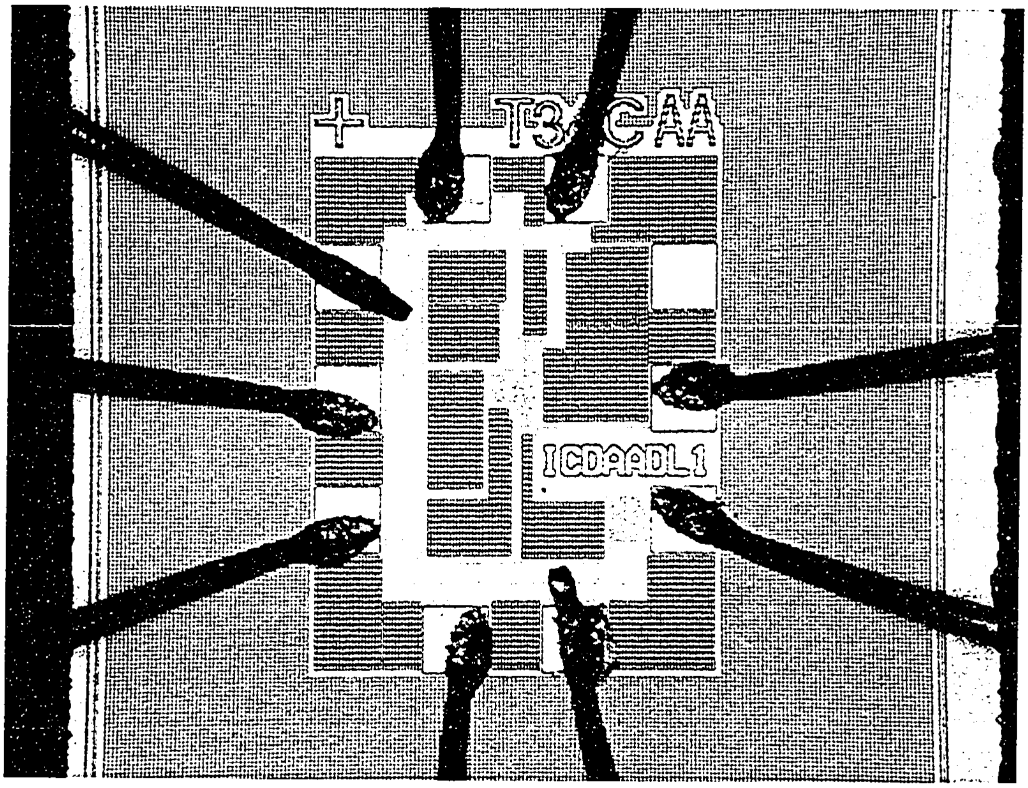


Figure 3.8: Die photograph of the prototype chip

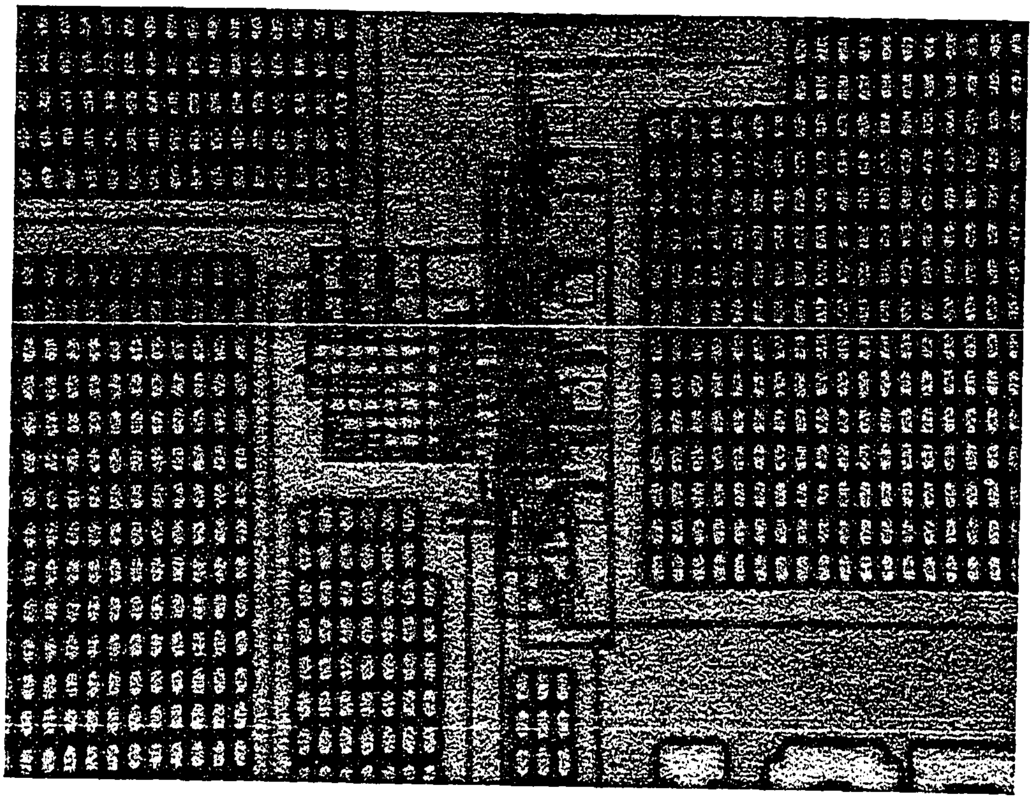


Figure 3.9: Die photograph of the circuits in the prototype chip

3.2 Testing prototype chip

The purposes of the prototype chip are to model the different pn junction photodiodes and to find the advantages and disadvantages of 3 transistor and 4 transistor pixel structures as well as N_{mos} and P_{mos} resets. A printed circuit board (PCB) was designed to connect the chip to a field programmable gate array (FPGA) and an ADC chip.

3.2.1 Test setup

The chip requires both analog and digital input signals for testing. The digital input signals are the reset and row select signals and they are generated by a FPGA. VHDL is used to program the FPGA. The analog input signals are a bias voltage of 0.7V for the current source transistor, and a reference voltage of half V_{DD} for the 4 transistor pixel structure. The test chip uses a supply voltage of V_{DD} = 3.3V. The voltage signals are generated by simple variable resistor circuits and light waves are generated by a halogen lamp with a 20W bulb. A neutral density filter is put on top of the chip and filters out 75% of light coming in, therefore only 25% of the light from the halogen lamp reaches the prototype chip. White light with different intensities are shone onto the chip and the corresponding output signals are collected. The collected output signals are converted to digital signals by the ADC and sent back to a PC for analysis and storage. The ADC chip is also controlled by the FPGA. Figure 3.10 illustrates the setup for testing the chip and Figure 3.11 shows the FPGA and the test board.

3.2.2 Test results

The power consumption of the chip is less than 3.3mW, but it was not possible to measure an accurate result. The current drawn by the chip is too small because there are only 8 pixels. The simulated power consumption by the core is about 0.9mW.

The output voltages of the 8 different pixels at different integration times are plotted in Figure 3.12. Pixels 5 and 6 have the lowest output voltages. At first, this looks like the two pixels convert more photons to current than the other pixels and have better or faster response. With closer inspection, pixels 7 and 8 are similar to pixels 5 and 6 except for the type of reset transistors. The reason why pixels 7 and 8 have higher output voltage is that the reset transistors for the two pixels are p

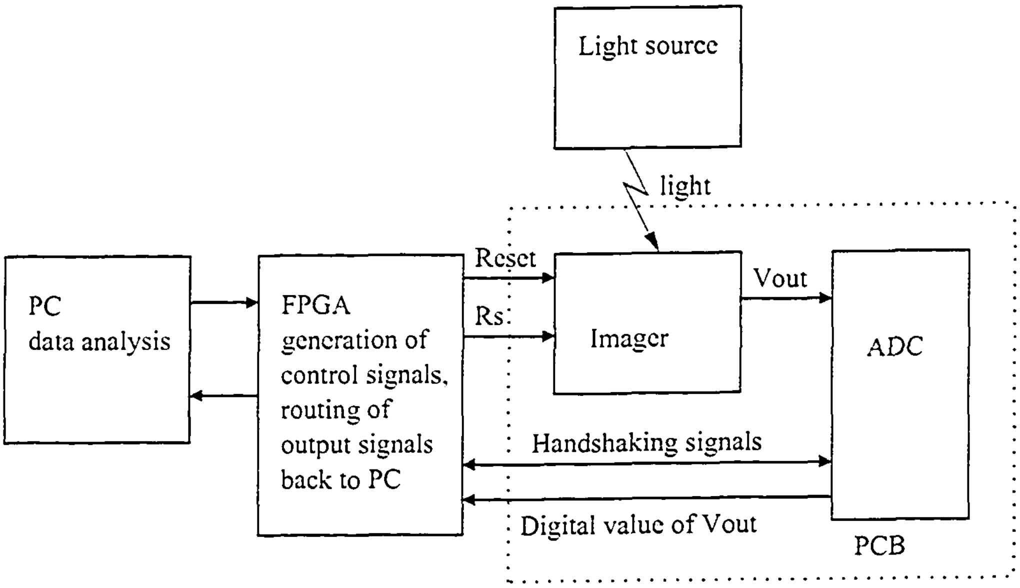


Figure 3.10: Setup for testing prototype chip

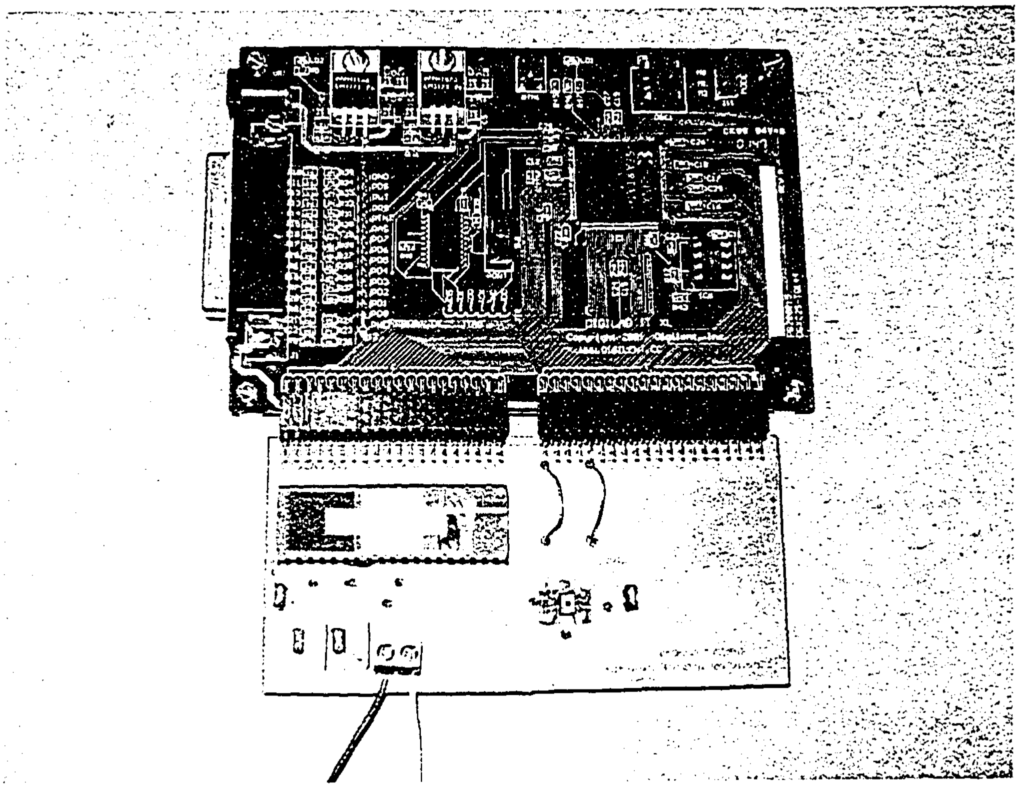


Figure 3.11: FPGA and test board with prototype chip

type and there is no V_t drop. From the experimental data, the difference in output voltage at 10μs between pixel 5 and 7 is 0.647V. This value is close to the value of V_t, which is around 0.6V. There is not much difference between pixels 1 and 2, pixels 3 and 4, pixels 5 and 6, and pixels 7 and 8. This means that for white light, n^+ - p_{sub} photodiodes and n_{well} - p_{sub} photodiodes have approximately the same response.

Generally, 4 transistor pixels have steeper slopes, which means these pixels generated more current assuming the capacitance was equal. The relationship between the slope of the curves, current and capacitance is:

$$\frac{dV}{dt} = \frac{I}{C} \tag{3.1}$$

where $\frac{dV}{dt}$ is the voltage change or the slope of the curve, C is the capacitance and I is the current. The equation shows that a steeper slope may be due to a higher current, lower capacitance or a combination of the two. Pixels 7 and 8 have steeper slopes compared to pixels 3 and 4, and pixels 5 and 6 also have steeper slopes compared to pixels 1 and 2. The slopes of pixels 3 and 4 are smaller than the rest of the pixels. A possible reason for this difference is that there is more parasitic capacitance in pixels 3 and 4 and this capacitance is mostly contributed by the drain of the P_{mos} reset transistor. With more capacitance in pixels 3 and 4, the voltage is discharged at slower rate and resulted in the smaller slopes. The photodiode capacitance in pixels 7 and 8 is only connected to the transistor M4 and separated from the drain of the P_{mos} reset and the amplifier transistor. Therefore, the capacitance in pixels 7 and 8 is smaller and the slopes are steeper. Also, the slopes of the 4 transistor pixel tend to decrease when the pixel voltage drops below 0.5V. The reason is that the transistor M4 is no longer operating in saturation region.

The results are similar in Figure 3.13 and 3.14 except that the response is over a longer integration time because the light source is at a further distance. Figure 3.15 shows the response of pixel 3 with light source at various distances. The slopes show that the pixel response is inversely proportional to the distance squared. The slope of light source at 15cm is about 4 times the slope of light source at 30cm and about 1.78 times the slope of light source at 20cm.

The pixels are also tested when there is no illumination. These test gives the time it takes for dark current to have an effect on the voltage level. The results are shown in Figure 3.16. For pixels 1, 2, 5, and 6, dark current starts to lower output voltage level at around 5ms. For pixels 7 and 8, the time is around 50ms. For pixels 3 and 4, the time is around 100ms. This tests show that n^+ - p_{sub} and n_{well} - p_{sub} photodiodes have the same amount of dark current. The pixels with PMOS reset and 3 transistors have less dark current than those with N_{mos} reset and 4 transistor pixels respectively. The estimated dark current values are shown in Table 3.2. The dark current values are calculated using Equation 3.1 by multiplying the slope with the photodiode capacitance. The capacitance is composed of photodiode capacitance calculated from listed values from CMC and parasitic capacitance.

Table 3.2: Estimated dark current values of the 8 pixels

Pixel	1	2	3	4	5	6	7	8
Dark current (fA)	198.1	233.9	54.9	64.9	588.6	694.9	232	275.2
Slope	4.75	4.75	1.29	1.29	14.24	14.24	5.63	5.63
Capacitance (fF)	41.75	49.3	42.45	50.18	41.35	48.81	41.18	48.85

According to the background theory of Chapter 2, there is no reason for a systematic difference in the photo/dark currents in the n^+ - p_{sub} photodiodes of pixels 1, 3, 5 and 7 or in the n_{well} - p_{sub} photodiodes of pixels 2, 4, 6 and 8 because the photodiode layouts are identical within these groups. Considering Equation 3.1, measured differences in $\frac{dV}{dt}$ are likely due to a net difference of parasitic capacitance and/or parasitic current in p-n junctions at the source/drain of transistors connected to the photodiodes. It is not easy to calculate the parasitic capacitance accurately from the process parameters provided by CMC alone.

3.3 Chapter conclusion

The tests results provide useful information on the various pixel designs. They give a better understanding on the pixel design tradeoffs. The 4 transistor pixels have better response, but the voltage range for linear operation is smaller. If short integration time is not required, then pixels 3 and 4 is a better choice because the voltage range for linear operation is bigger. Also, pixels 3 and 4 have the least amount of dark current. The results showed that with higher light intensity (light source at a closer distance), the voltage will decrease at a faster rate. This is consistent with Equation 2.8 from chapter 2 where current density is proportional to incident photon flux. The next chapter will describe two types of signal processing for incorporating into image sensors.

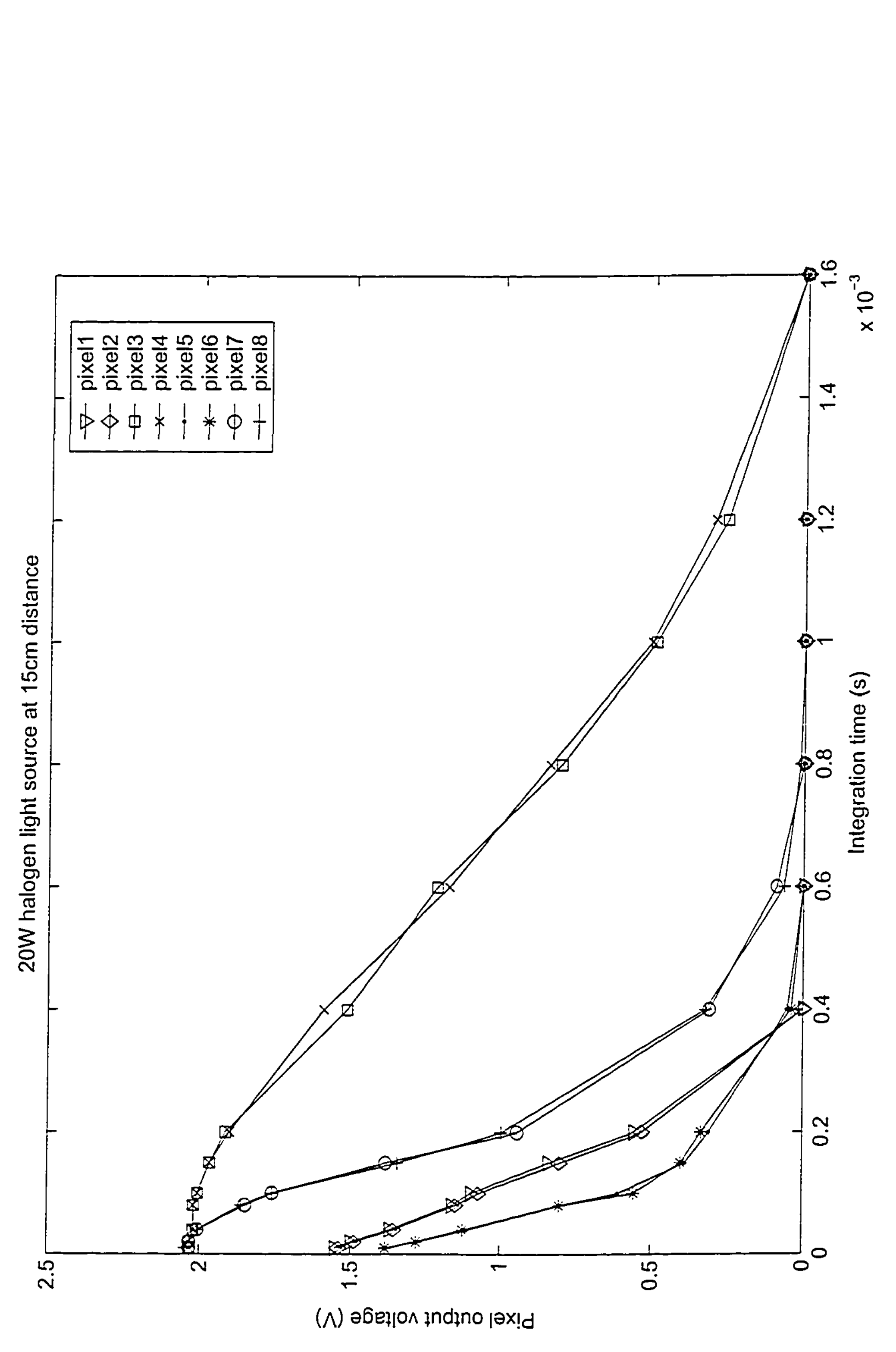


Figure 3.12: Output voltage vs integration time with light source at 15cm

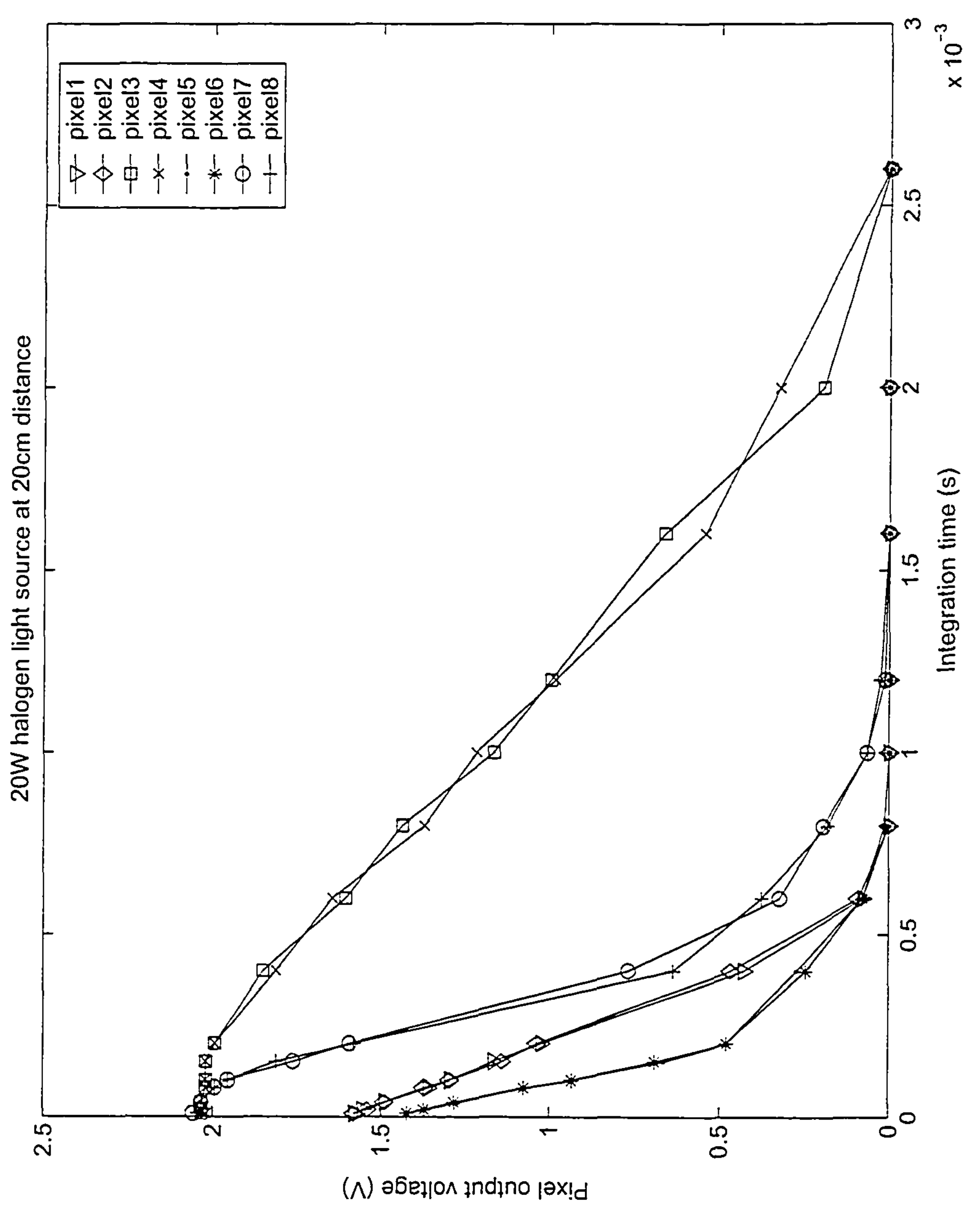


Figure 3.13: Output voltage vs integration time with light source at 20cm

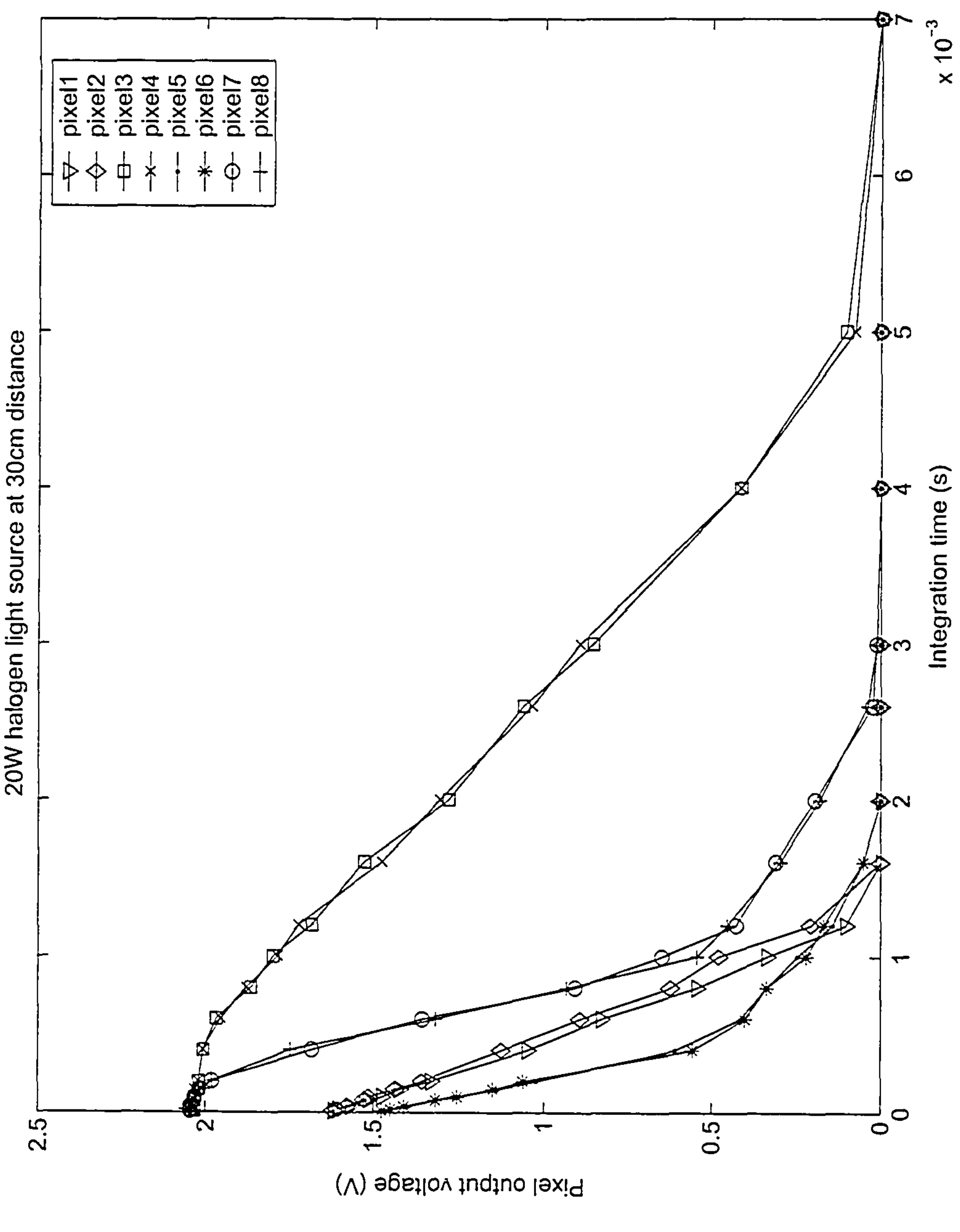


Figure 3.14: Output voltage vs integration time with light source at 30cm

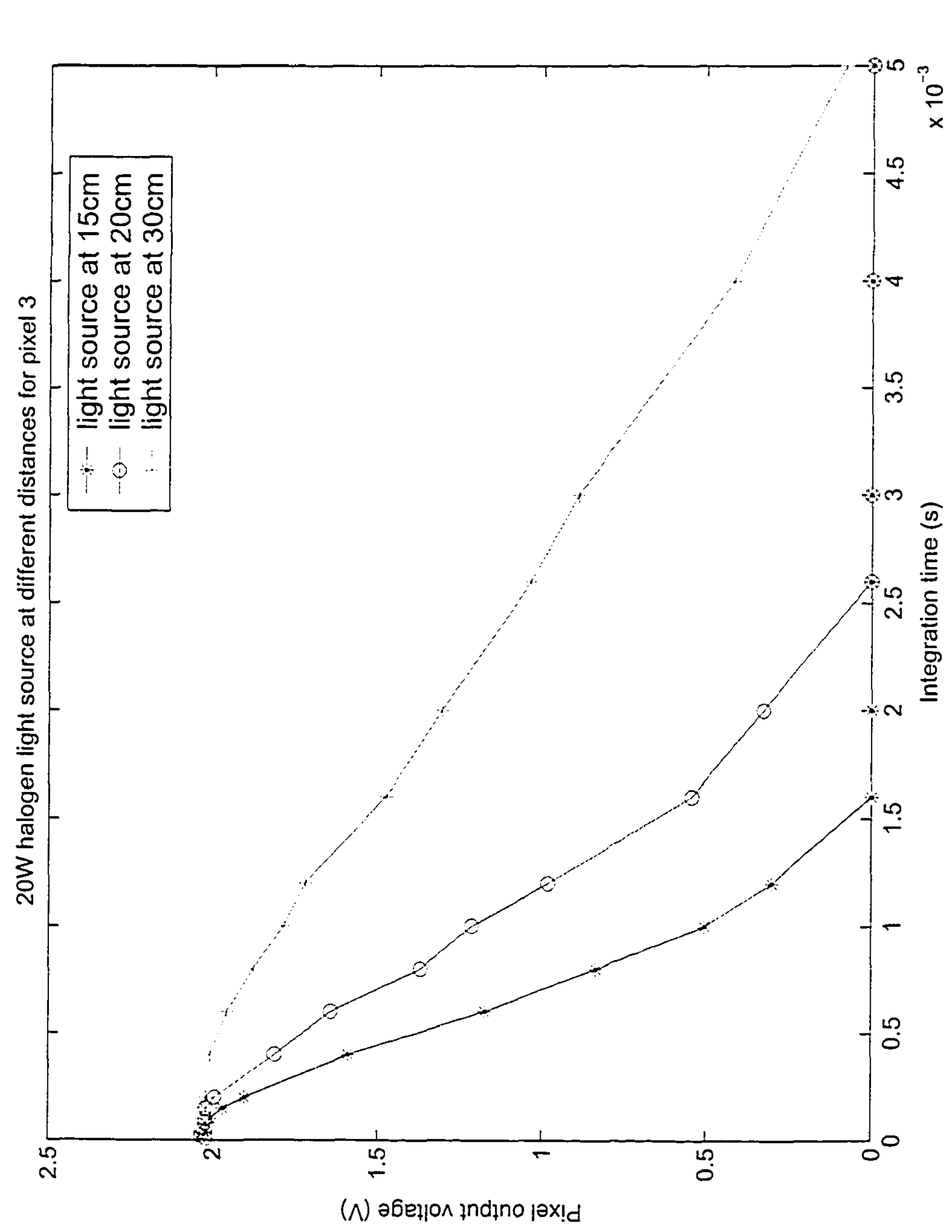


Figure 3.15: Output voltage vs integration time of pixel 3 with light source at various distances

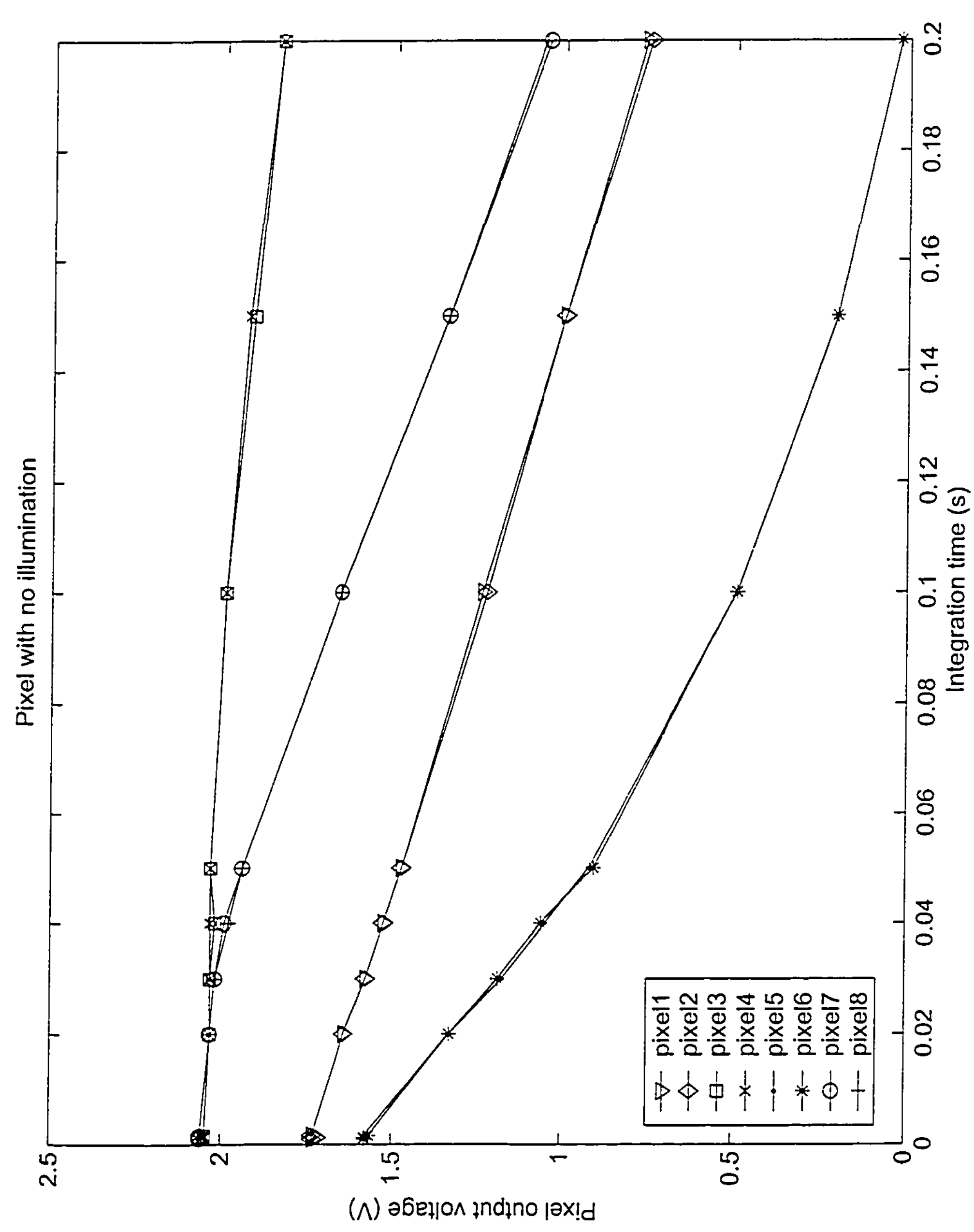


Figure 3.16: Output voltage vs integration time with no illumination

Chapter 4

Background on signal processing

Chapter 3 gave some experimental results on several CMOS pixel designs. Computational CMOS image sensors were discussed previously and they have the advantages of supporting parallel processing and reduced bandwidth requirements. Stochastic computation and pulse width modulation (PWM) are methods to encode a signal from voltage domain to time domain and they are alternatives to traditional computation circuits. Stochastic computation and PWM provide many benefits over other computing techniques such as very low computation hardware area and simple hardware implementations. This section shifts the focus to signal processing and statistics to give a better understanding of how stochastic computation and PWM work, and how they might be applied to computational imagers.

4.1 Stochastic computation

4.1.1 Stochastic representation and computation

Stochastic computation was developed in the 1960s to combat the high cost of computational elements. Digital computation required a large number of transistors and in the 1960s, transistors were expensive and difficult to integrate. By sacrificing accuracy, stochastic computation approximates the computations with less hardware by serializing the computation process. The advantages of stochastic computation over conventional computation are very low computation hardware area, fault tolerance, simple serial communications per signal, simple hardware implementations enabling very high clock rates, and capability to trade off computation time and accuracy without hardware changes.

Stochastic computation is achieved through representation of data by the probability that a logic level will be ON or OFF during a clock period [11]. In binary computation the logic levels representing data change deterministically from value to value, and if the computation is repeated the same sequence of logic levels will occur. In stochastic computation arithmetic operations are performed by virtue of the completely random and uncorrelated nature of the logic levels representing data, and only the probability that a logic level will be ON or OFF is determined: its actual value is a chance event which cannot be predicted, and repetition of a computation will give rise to a different sequence of logic levels.

A stochastic stream is a sequence of bits, in which the information is contained in the primary statistic of the bit stream <X> where <X> = $\frac{N}{N_{max}}$ and N is the number to be represented and N_{max} is the maximum value that N can be. For example, Table 4.1 shows some samples of stochastic sequences with <X> = 0.5, N = 10 and N_{max} = 20.

Table 4.1: Examples of stochastic sequences for $< X >$ = 0.5

a	01001001110110011010
b	01110001100010101011
c	11011001110110001000
d	00010111101110010010

Any sequence may occur, but the proportion of ON levels in a large sample will have a binomial distribution with a mean of <X> or 0.5 in Table 4.1. Observation of the information carried on a given signal line is therefore a probabilistic process itself [4]. Shown in Figure 4.1, multiplication can be done by using a simple AND gate and weighted sum or averaging can be calculated by using a multiplexer (MUX). In conventional signal processing, multiplication is the bottleneck and requires a large area in an IC. By using stochastic signal processing, less resources are required. Table 4.2 shows the major operations that can be performed by stochastic computation [11].

Table 4.2: Major operations that can be perform by stochastic computation

Circuit components	Function in stochastic encoding
AND gate	multiplication
AND gate and D flip-flop	squaring
MUX	averaging or weighted addition
AND gate and up/down counter	dividing

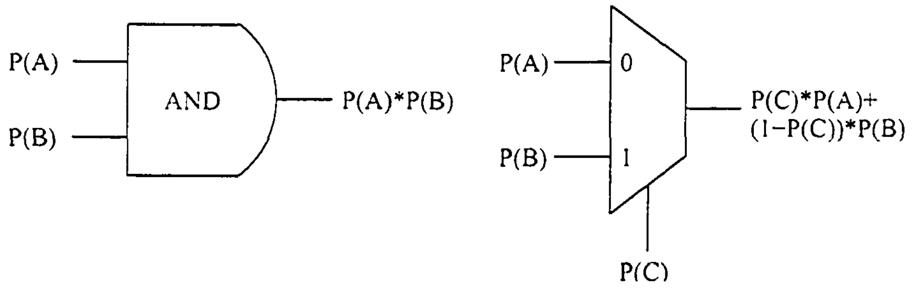


Figure 4.1: Using digital logic gates to perform different functions on stochastic streams

4.1.2 Examples of stochastic computation

Examples of stochastic computation are shown in Table 4.3. In Table 4.3, two stochastic streams are presented, stream A with a probability of 0.5 and stream B with a probability of 0.4. Two different sequences are sampled from each stream and the result of these sequences going through an AND gate are listed. A third stream C with a probability of 0.5 is also presented in the table for the averaging function. The sequences sampled from this stream is fed into the select line of the MUX in Figure 4.1. Using a probability of 0.5 for the select line, the MUX will average the two probabilities of A and B.

Table 4.3: Examples of stochastic computation

	probability	10 bit sequence	20 bit sequence
A	0.5	1101001010	11001010100011001110
B	0.4	1010100100	10010011100000110010
A*B	P(A)*P(B)	1000000000	10000010100000000010
A*B	0.5*0.4 = 0.2	A*B = 0.1	A*B = 0.2
C	0.5	0110101010	01101011000001111001
average of A & B	P(A)*P(C)+ (1-P(C))*P(B)	1011100000	10000011100010110110
average of A & B	0.5*0.5+ 0.5*0.4 = 0.45	average = 0.4	average = 0.45

4.1.3 Law of Large Numbers and central limit theorem

Stochastic computation involves randomly sampling a signal. Therefore, it is useful to understand the Law of Large Numbers [21]. The Law of Large Numbers states that if an event A with probability *P(A)* $= p$ occurs k times in n trials, then k

$\approx np$. The approximation $k \approx np$ means that the ratio $\frac{k}{n}$ is close to p in the sense that, for any $\epsilon > 0$, the probability that $\mid \frac{k}{n} - p \mid < \epsilon$ tends to 1 as $n \rightarrow \infty$. This law is useful because it can be used to determine the number of samples needed to reach a certain reliable estimate. Consider a population having mean μ and finite standard deviation σ. Let $\bar{x}$ represent the mean of n independent random observations from this population. The central limit theorem states that as n becomes large, the sampling distribution of $\bar{x}$ tends toward a normal distribution with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$. This theorem permits us to approximate the sampling distribution for $\bar{x}$ by an appropriate normal curve regardless of the form of the population frequency distribution. Equation 4.1 is used to estimate the mean μ based on a sample with size n. $\bar{x}$ is the sample mean, $z_{\frac{\alpha}{2}}$ is the z-value beyond which the area under the normal distribution curve is equal to $\frac{\alpha}{2}$, σ is the standard deviation, n is the sample size, and $1 - \alpha$ is the confidence level.

$$P\left(\bar{x} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha \tag{4.1}$$

For example, with $\bar{x} = 50$, $\sigma = 12$, and $n = 144$, we want to find the confidence interval with 90% confidence level. Using Equation 4.1, the confidence level of 90% is $1 - \alpha = 0.9$, $\alpha = 0.1$ and $\frac{\alpha}{2} = 0.05$. $z_{\frac{\alpha}{2}} = z_{0.05} = 1.645$, $\frac{\sigma}{\sqrt{n}} = \frac{12}{\sqrt{144}} = 1$, and $\mu = 50 \pm 1.645$. Therefore, the confidence interval for this group of data with 90% confidence level will be [48.355, 51.645]. With the same sample mean and standard deviation, if the sample size is increased to 576 and 2304, the confidence interval with 90% confidence will be $\mu = 50 \pm 0.8225$ and $\mu = 50 \pm 0.41125$ respectively. The confidence interval depends on the number of samples, which is the $\frac{1}{\sqrt{n}}$ term, therefore it will decrease with the increase of the sample size. When the sample size increases, the probability of this group of data will therefore fall in a more narrow range and therefore, be more accurate.

For a confidence interval of population proportion, Equation 4.2 is used. $\hat{p}$ is the sample proportion, $\hat{q} = 1 - \hat{p}$, n is the sample size, and p is the population proportion.

$$p = \hat{p} \pm z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}\hat{q}}{n}} \tag{4.2}$$

For example, 5500 students were sampled, and 2849 have a cell phone, we want to estimate the proportion of students in that city with cell phone with 90% confidence level. $\hat{p} = \frac{2849}{5500} = 0.518$, $\hat{q} = 1 - 0.518 = 0.482$, $z_{\frac{a}{2}} = 1.645$, $p = 0.518 \pm 0.01108$. Assuming no sampling bias, this means between 50.692% and 52.908% of students have a cell phone with 90% confidence.

Equation 4.2 is useful to calculate the number of samples to ensure results with a certain confidence level. For a stochastic stream of bits with a 90% probability that each bit will be a 1, if we want a confidence interval of 0.89 - 0.91 with 90% confidence, $\hat{p} = 0.9$, $\hat{q} = 0.1$, $0.01 = 1.65\sqrt{\frac{0.9*0.1}{n}}$, and n = 2450. Table 4.4 lists the number of samples needed to achieve different confidence intervals with 90% confidence. The table shows that in order to reach a high confidence interval, a large amount of bits are required.

Table 4.4: Number of samples to achieve different confidence intervals with 90% confidence

$\hat{p}$	number of samples with confidence interval $\hat{p} \pm 0.05$	number of samples with confidence interval $\hat{p} \pm 0.02$	number of samples with confidence interval $\hat{p} \pm 0.01$
0.1	98	612	2450
0.5	272	1701	6806
0.9	98	612	2450

Higher precision computation can be done by observing the sequence of bits for a longer time in order to allow a lower variance estimate of the average to be made. By observing the sequence of bits for a longer time, the number of samples will be bigger. The Law of Large Numbers states that the probability that | $\frac{k}{n}$ - p | $< \epsilon$ tends to 1 as $n \rightarrow \infty$. This means that with more samples, the results of stochastic computation will be more accurate and this is illustrated in Table 4.3 by comparing streams of 10 bits and 20 bits. If we observe a sequence of N logic levels and k of them are ON, then the estimated generating probability is:

$$\hat{p} = \frac{k}{N} \tag{4.3}$$

The sampling distribution of the variable k is binomial, and hence the standard deviation of the estimated probability $\hat{p}$ from the true probability p is:

$$\sigma(\hat{p}) = \sqrt{\frac{p*(1-p)}{N}} \tag{4.4}$$

For Table 4.3, the standard deviation of a 10 bit sequence with probability $p =$

0.5 is $\sigma(\hat{p}) = \sqrt{\frac{0.5*0.5}{10}} = 0.158$. For a 20 bit sequence with $p = 0.5$, $\sigma(\hat{p}) = 0.112$. Therefore, with more bits sample, the standard deviation will get smaller, which means it is closer to the true probability.

4.1.4 Comparing stochastic and binary radix integer systems

In [4], a stochastic system is compared with a binary radix integer (BRI) system. In serial communication of data, BRI representation is exponentially faster than unary stochastic representations. Data with a resolution of M bits requires a communication time M for BRI, but a time 2^M for the unary stochastic case. The 2^M time does not guarantee the exact information is sent because the stochastic system is a random process. The area-time product is a measure of the energy of a computation with the assumption that there is a constant power dissipation per unit area. Brown and Card [4] calculated the area-time product. For $M < 10$ bits, the stochastic system has better performance than the BRI system. For $M > 10$ bits, the BRI system performs better.

If precision is more important, then it would be better to use the BRI system. In a stochastic system with 8 bit resolution, using Equation 4.2, 2^M or 256 cycles will only give a confidence interval of [0.45, 0.55] for $\hat{p} = 0.5$. If higher precision is needed, then the time needed for communication will be longer. The observation of long bit streams will delay the overall system.

Another important consideration is the comparison of error tolerance between the two systems. In the worst case scenario for a single bit error, the maximum error in the BRI case is when the error bit is in the most significant bit (MSB). The error is $\frac{2^{M-1}}{2^M} = \frac{1}{2}$ of the maximum value. The maximum error in the stochastic case is only $\frac{1}{2^M}$ of the maximum value and this means that the stochastic systems have much better error tolerance than BRI systems.

4.1.5 Generating stochastic sequences

For encoding a binary digital number to a stochastic sequence, a random number is compared with the digital number stored in a register. The top circuit in Figure 4.2 is an example of a digital-to-stochastic converter. This random number generator requires digital noise bits where the probability of each bit being a one is $\frac{1}{2}$. Gaines [11] describes a number of potential sources of digital noise. One common generator is the linear feedback shift register (LFSR). The LFSR consists of shift registers and

XOR gates.

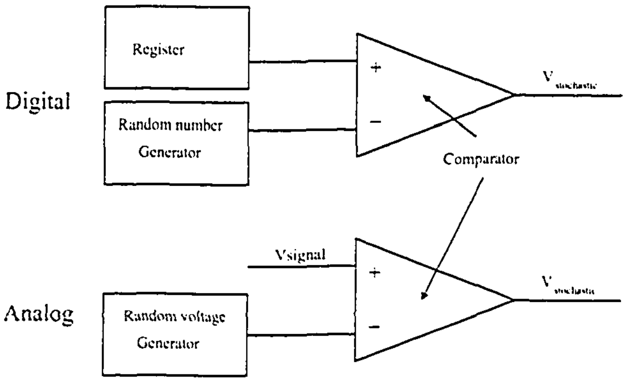


Figure 4.2: Circuit to generate stochastic signals

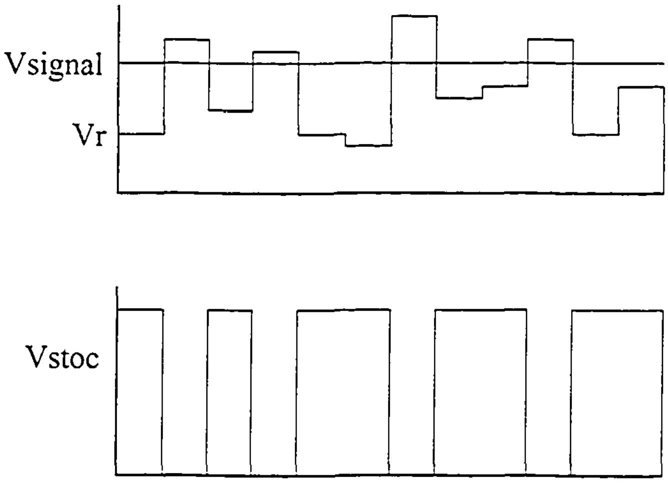


Figure 4.3: Waveform of generating stochastic signals

The design of the proposed image sensors require that the analog output signal voltage from each pixel be converted to a stochastic sequence. Implementing an LFSR and then converting the random number to an analog voltage will take up too much space. Therefore, the designs for digital-to-stochastic converters are not applicable in this thesis. To generate stochastic sequences from an analog signal voltage, a random voltage generator such as a noise diode or resistor is used as a noise source [11]. The purpose of this noise source is to generate a random voltage. V_r, and this voltage is compared with the signal voltage, V_{signal}. If V_r is less than or equal to V_{signal}, the output of the comparator is set to high. Otherwise, V_{stoc} is set to low. Figure 4.3 shows a waveform of a particular conversion and the bottom circuit in Figure 4.2 is an example of a analog-to-stochastic converter. There are many analog noise generators available but these circuits are complex and require a large amount of space. Also, if high precision is needed, then we will need to observe a large amount of bits. Therefore, it is not feasible to implement stochastic computation units in pixels. Instead, we will explore the use of PWM coding for signal processing. PWM signal processing is discussed in the next section.

4.2 Pulse width modulation

Aside from stochastic computation, PWM is another way of encoding an analog signal level to a single-bit serial digital signal. In a PWM signal, the duty cycle of a square wave is proportional to the analog signal level it is encoding. The duty cycle is the percentage of a period that the signal is ON. For example, if the maximum voltage for a signal is 5V, the duty cycle of a 2.5V signal will be 50%. See Figure 4.4 for an example encoding. Any analog value between 0V and the maximum voltage can be encoded with PWM. If the analog signal is negative, then the duty cycle will be 0% and in the opposite end, if the signal is over the maximum voltage, then the duty cycle will be 100%.

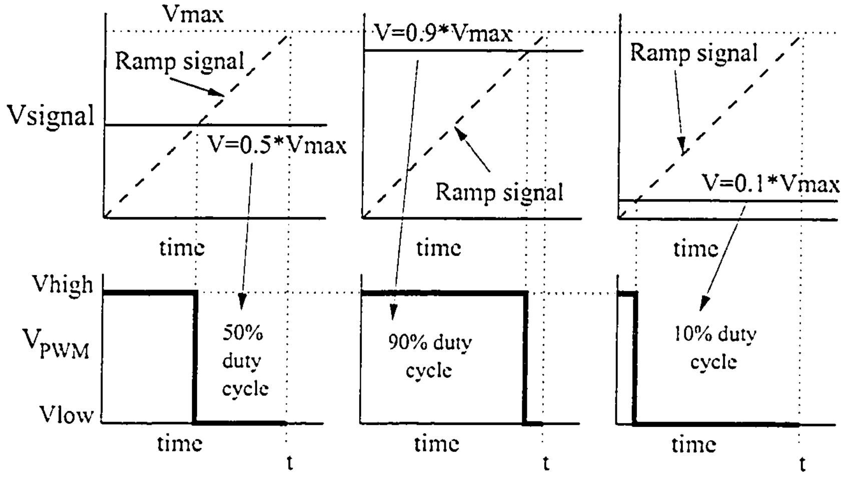


Figure 4.4: Different analog signal levels and equivalent PWM signals and duty cycles

One simple way to generate a PWM signal is to use a comparator and compare the analog signal to a ramp signal. In Figure 4.5, the comparator will output the ON voltage or V_{high} (in Figure 4.4) when the analog signal from the APS pixel V_{out}

(in Figure 2.5) is larger than the ramp signal. To generate a ramp signal, a current source is used to charge up a capacitor at a constant rate. If the slope of the ramp function is variable, then finer increments of the input signal can be encoded, which increases the dynamic range. Therefore, the dynamic range can be maintained even in low lighting conditions by using a ramp function with a smaller slope. A reset switch is also available to reset the circuit and start another cycle. The design of the comparator and ramp generator will be discussed in the next chapter.

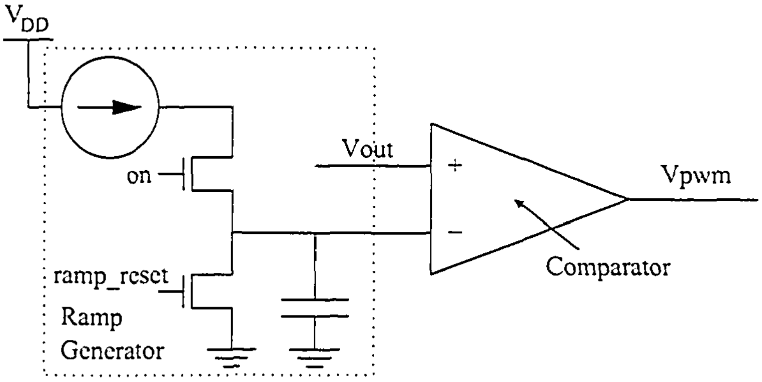
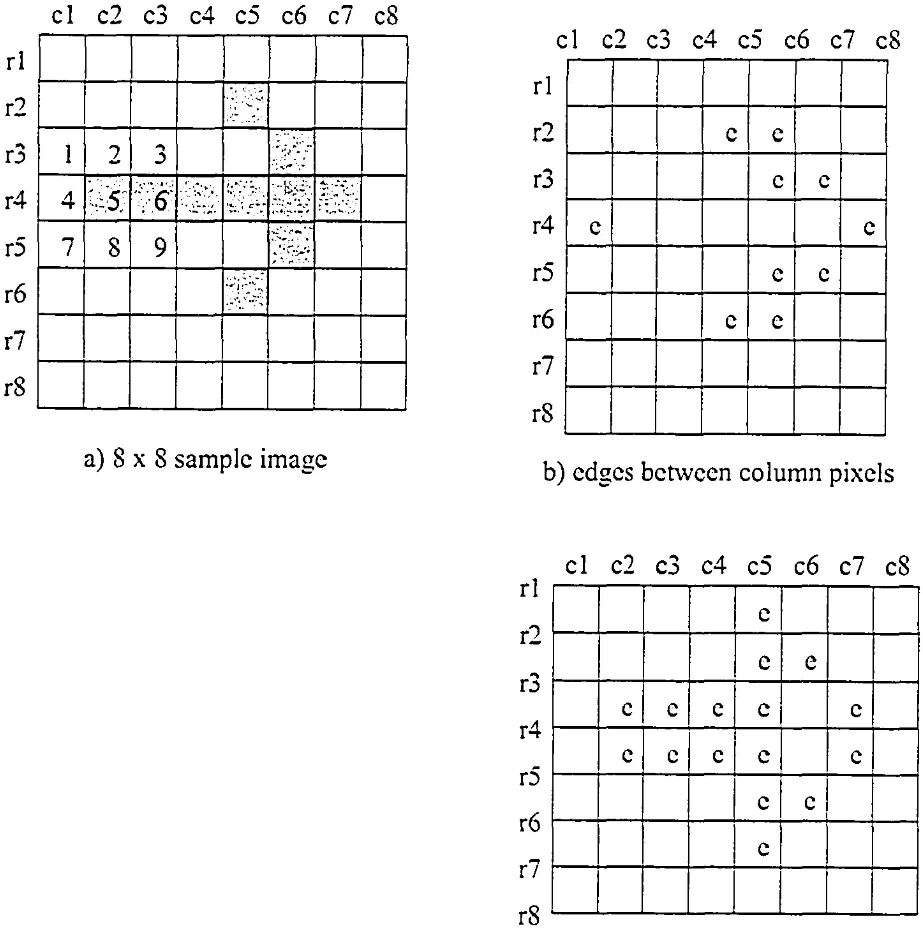


Figure 4.5: Circuit to generate PWM signals

Similarly to stochastic computation, simple digital logic can be used to perform different functions on PWM signals. For example, if the duty cycles of two PWM signals both start at t=0, then an OR gate can be used to find the maximum of the two signals. With the same assumption, an AND gate can be used to find the minimum and a XOR gate can find the difference between two PWM signals. The difference between two PWM signals can be used for edge detection applications. There is an edge when two neighboring pixels have a large difference in light level.

To understand what defines an edge, an example is shown in Figure 4.6. Figure 4.6a shows a 64 pixel sample image with some dark and bright pixels. In the 3x3 squares numbered 1 to 9, there are 5 edges. There is one vertical edge between pixels 4 and 5. There are also four horizontal edges and their positions are between pixels 2 and 5, pixels 3 and 6, pixels 5 and 8, and pixels 6 and 9. Figure 4.6b shows the edges between column pixels (vertical edges). If there is an e in the box, then there is an edge. The vertical edge between pixels 4 and 5 is listed in the box r4 between c1 and c2. The edges between row pixels (horizontal edges) are listed in Figure 4.6c.



	c1	c2	c3	c4	c5	c6	c7	c8
r1								
r2					▨			
r3	1	2	3			▨		
r4	4	▨ 5	▨ 6	▨	▨	▨	▨	
r5	7	8	9			▨		
r6					▨			
r7								
r8								

a) 8 x 8 sample image

r1							
r2				c	c		
r3					c	c	
r4	c						c
r5					c	c	
r6				c	c		
r7							
r8							

b) edges between column pixels

	c1	c2	c3	c4	c5	c6	c7	c8
r1–r2					c			
r2–r3					c	c		
r3–r4		c	c	c	c		c	
r4–r5		c	c	c	c		c	
r5–r6					c	c		
r6–r7					c			
r7–r8								

c) edges between row pixels

Figure 4.6: A 8x8 pixel sample image and edges

Figure 4.7 shows various waveforms resulting from using different logic gates to perform the functions described in Table 4.5. The corresponding functions in stochastic computation are also described in the table. By measuring the number of clock cycles of the A XOR B or C XOR D signal when it is high, the XOR function can be used for edge detection. Using a comparator, if the A XOR B signal is ON for longer than a certain reference time period, say half the time period of a full cycle, then there is an edge between pixels A and B. Whereas in the C XOR D signal, there is no edge between pixels C and D. The determination of an edge depends on the reference value. If a smaller reference time period is chosen and it is less than the C XOR D signal, then there is an edge between pixels C and D.

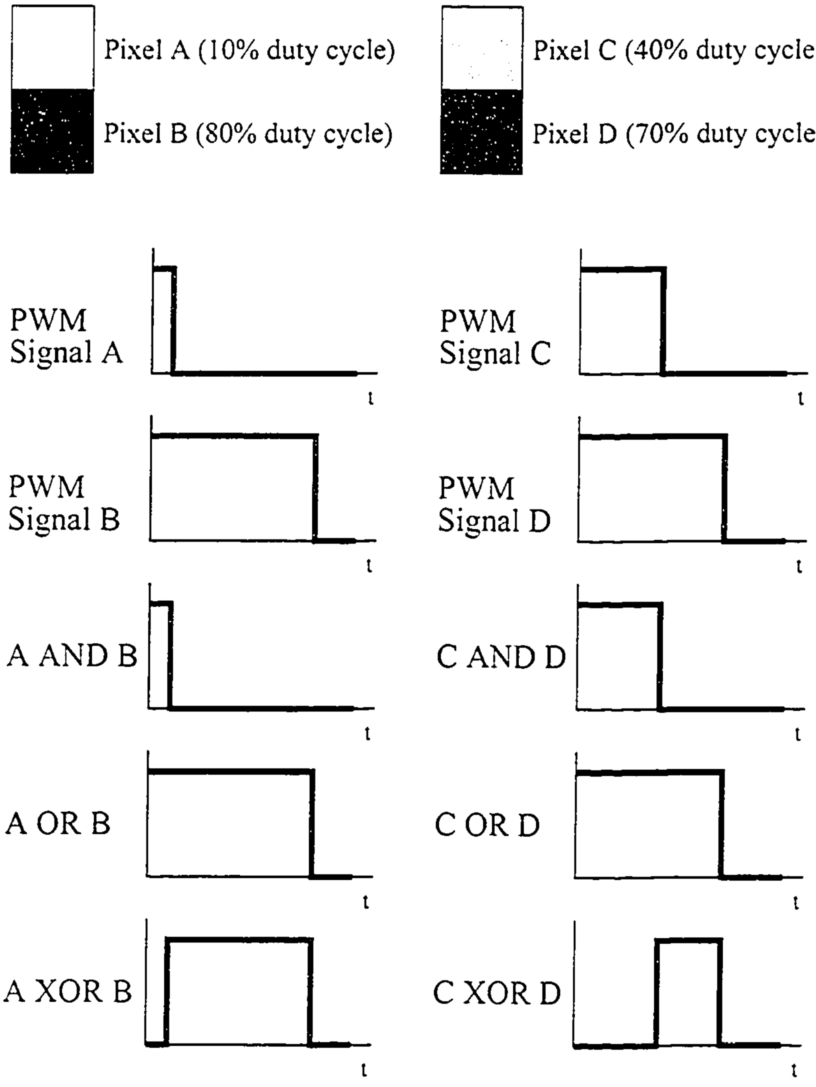


Figure 4.7: PWM functions using digital logic gates

Table 4.5: Logic gates and their function in PWM and stochastic encoding

Logic gate	Function in PWM	Function in stochastic encoding
And	minimum	multiplication P(A)*P(B)
Or	maximum	union P(A) + P(B) - P(A)*P(B)
Xor	difference	P(A)(1 - P(B)) + P(B)(1 - P(A))

4.3 Chapter conclusion

In this chapter, we discussed stochastic and PWM computations and examples are shown on how these methods can be applied in signal processing. We also discussed the Law of Large Numbers and the central limit theorem, stating that if the number of samples increases, the accuracy of the estimated mean will increase and will follow a normal distribution. In the next chapter, we will talk about combining PWM signal processing circuits with sensing circuits to form a computational image sensor.

Chapter 5

PWM computation in CMOS image sensors

The previous chapter gives an overview on how stochastic and PWM computations can be alternative methods of signal processing that can be incorporated into computational image sensors. In this chapter, the details of our proposed pixel design are described. We will discuss the image sensor operation and the advantages of this design.

5.1 Proposed structure

Stochastic computation can be applied to the signal processing on CMOS image sensors [24]. Another method to perform signal processing is to employ pixel level PWM signal processing. Nagata *et. al.* reported a smart CMOS image sensor with pixel level PWM signal processing [33]. This PWM signal CMOS image sensor performed block averaging and 2D projection with PWM signal addition techniques based on switched current integration and charge packet counting.

A CMOS image sensor with PWM signal processing circuits placed near the pixels is explored in this thesis. With signal processing at the pixel level, system bandwidth can be reduced by preprocessing the information and outputting just the important information for a particular application, instead of transferring the whole information. This proposed image sensor performs edge detection functions using single logic gates, a concept similar to [13] and [24]. The difference in this proposed image sensor is that the voltage signal of each pixel is encoded to a PWM signal. instead of a stochastic signal. Also, this proposed image sensor performs subtraction using a simple logic gate, unlike the current integration technique used by [33] to perform addition. One of the advantages of this proposed imager is the simplicity of the analog-to-PWM circuits. Another advantage is that the PWM signal processing circuits use very few components, mainly the XOR gate and the integrator circuit to perform edge detection. These simple circuits enable us to embed these circuits into the pixel array. PWM is used to encode the pixel output from the voltage domain to the time domain for signal processing. The new proposed image sensors have a new pixel structure that consists of 2 blocks as illustrated in Figure 5.1. This will add overhead and increase the area of each pixel. Also, there are signal processing circuits for every 2 pixels.

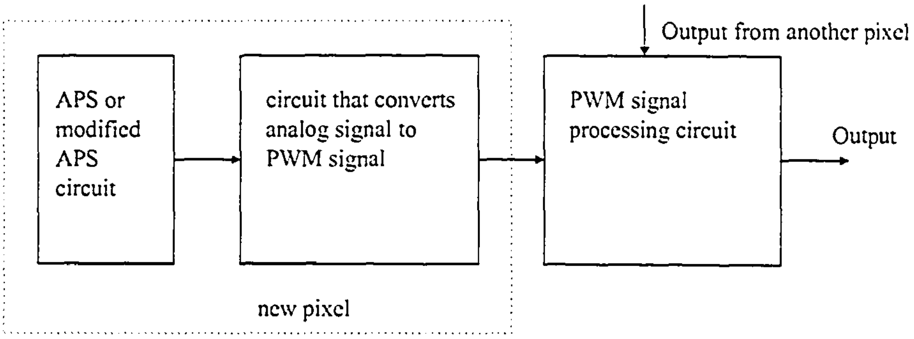

Output from another pixel
APS or modified APS circuit
circuit that converts analog signal to PWM signal
PWM signal processing circuit
Output
new pixel

Figure 5.1: Proposed computational CMOS APS block diagram

5.2 Imager Architecture

Figure 5.2 shows the architecture of the proposed imager. In each pixel, instead of using the standard 3 transistor APS pixel, modifications are required. The amplifier transistor and row select transistor are taken out and a comparator is put in and the pixel is shown in Figure 5.3. A PMOS reset transistor is used to avoid V_t mismatch and allow the node N1 to reach V_{DD} when the pixel is reset. The comparator consists of a differential gain stage and a single-ended gain stage, followed by an inverter, as shown in Figure 5.6. This design was used in [18] for high-speed operation. The transistor sizes used for the comparator are minimum sizes, which is $0.8\mu m$ for width and $0.35\mu m$ for length. From simulation results, the offset voltage is 33mV and the maximum operating frequency is 227MHz. The ramp generator provided a ramp voltage that is connected to each pixel's comparator inverting input, V_-. The noninverting input, V_+, is connected to the node N1. The output of each pixel is a PWM signal, which is fed to neighboring processing circuits. Each pixel has a size of 23μm by 18μm containing 10 transistors and a photodiode. The fill factor of each pixel is about 14%. The layout of the proposed pixel is shown in Figure 5.4. Also, the layout of a block of 4 pixels is shown in Figure 5.5.

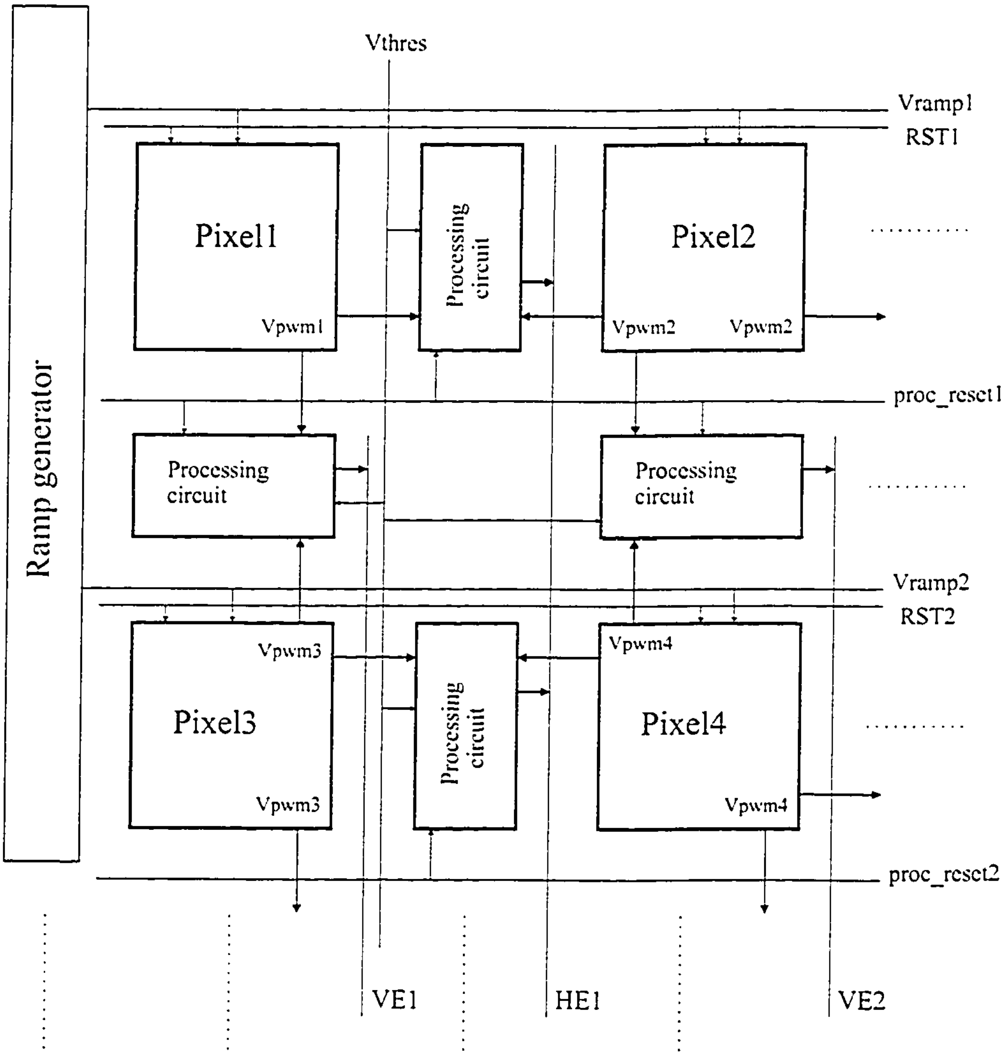


Figure 5.2: Proposed pixel-level architecture

Each processing circuit contains an XOR gate, an integrator circuit and a comparator. The schematic and layout are shown in Figures 5.7 and 5.4 respectively. The size of the processing circuit is 33μm by 11μm and it contains 19 transistors and a capacitor. The processing circuit calculates the difference of the two PWM signals and checks for an edge. If there is an edge, then a flag bit is set to 1. The next section talks about the edge detection circuit in more detail. Also, the ramp function used for encoding the pixel output is programmable, meaning the slope of

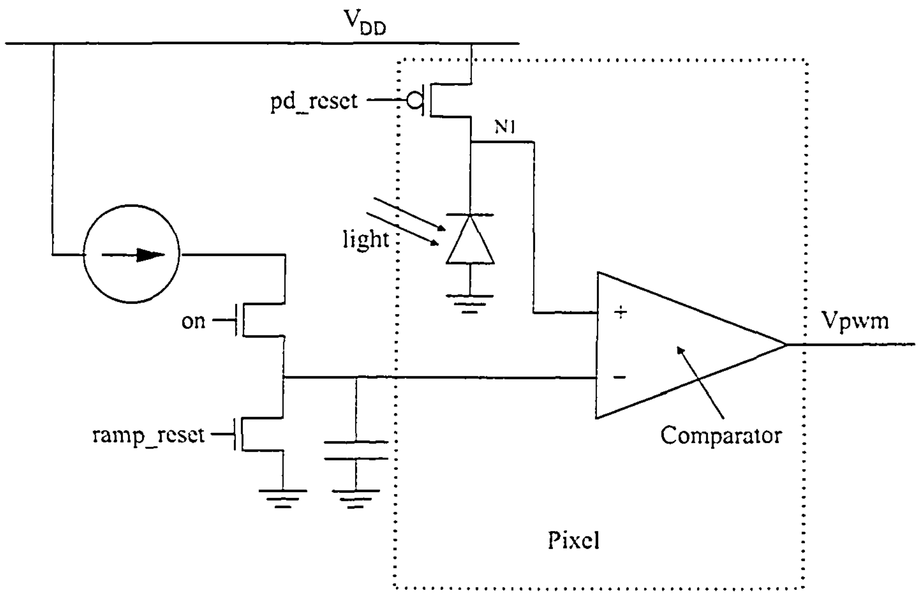


Figure 5.3: Proposed imager pixel schematic

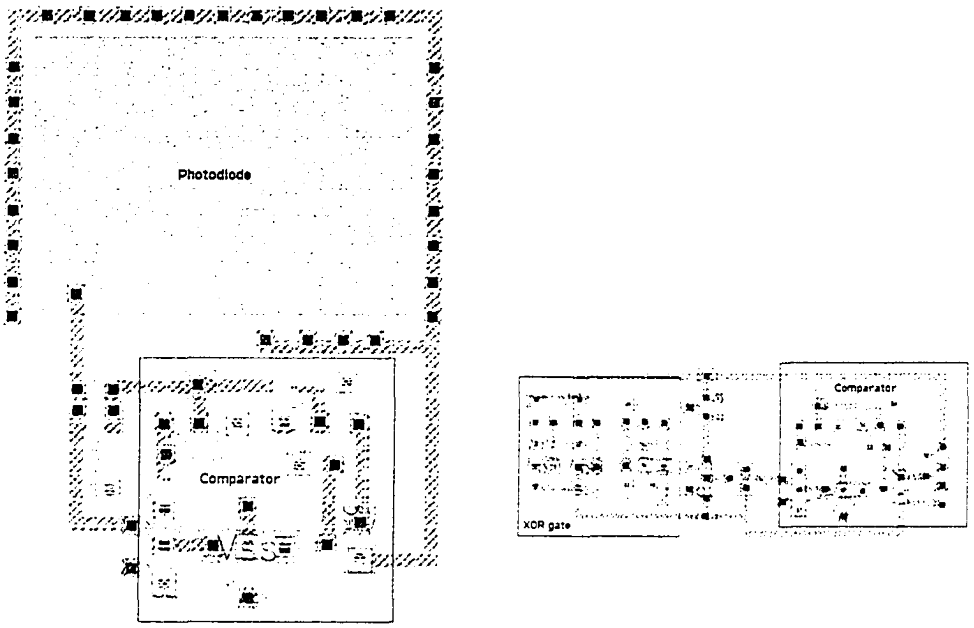


Figure 5.4: Proposed imager pixel and processing circuit layouts

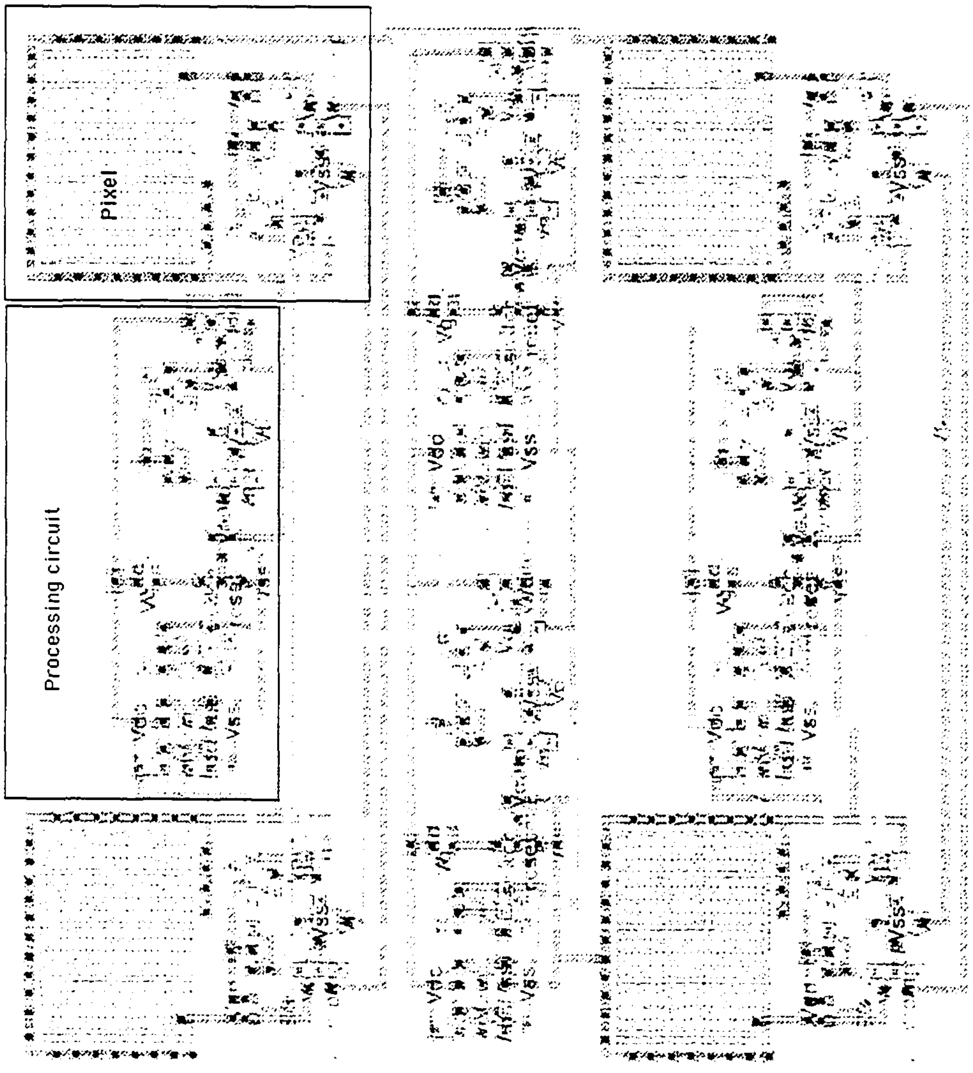


Figure 5.5: Layout of a 2x2 pixel

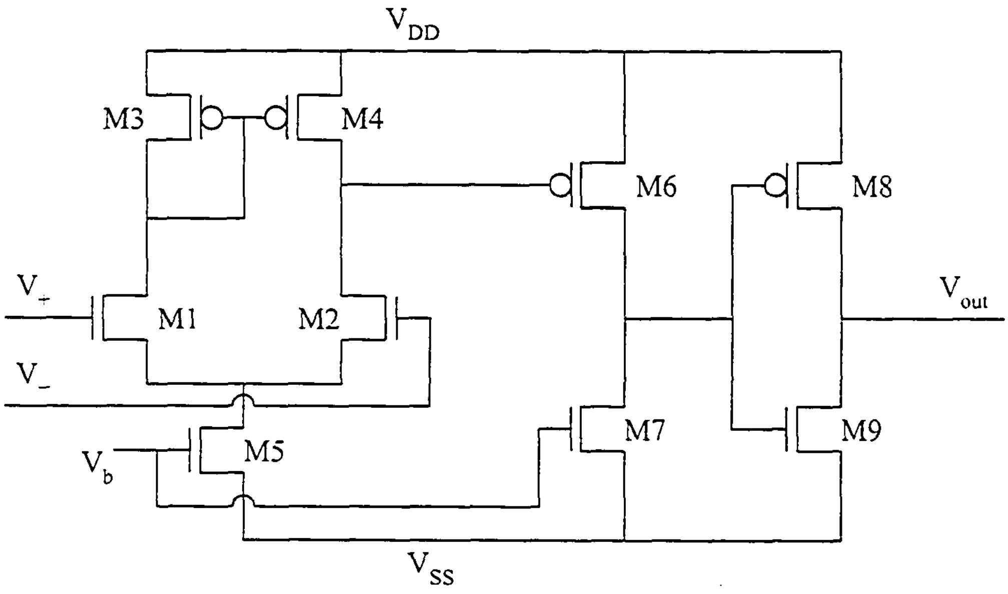


Figure 5.6: Comparator circuit

the ramp can be adjusted. By adjusting the ramp function, adaptive dynamic ranges are possible for varying lighting conditions. This architecture can detect horizontal and vertical edges.

5.2.1 Edge detection circuit

The circuit shown in Figure 5.7 is used for edge detection. An integrator circuit integrates the output of the XOR gate, analogous to counting the amount of time when V_{pwm1} XOR V_{pwm2} is ON. $V_{threshold}$ is the reference voltage to compare with the output of integrator circuit. Using a comparator, if the integrator output is larger than $V_{threshold}$ than there is an edge. Figure 5.8 shows the timing diagram of the edge detection circuit. V_{pwm1}-V_{pwm2}, the integrator output for pixel 1 and pixel 2, is larger than $V_{threshold}$, which means there is an edge between pixel 1 and pixel 2. Therefore, V_{edge} for V_{pwm1}-V_{pwm2} will output a 1. If the difference of two pixels is below $V_{threshold}$, like V_{pwm2}-V_{pwm3}, then there is no edge between pixel 2 and pixel 3.

5.2.2 Readout

If the flag bit, V_{edge}, is set to 1, then there is an edge between the two pixels. The voltage V_{diff} in Figure 5.7 is the edge information, i.e, the result of one pixel voltage subtracted from the other. The flag will enable an N_{mos} switch to pass V_{diff} to a

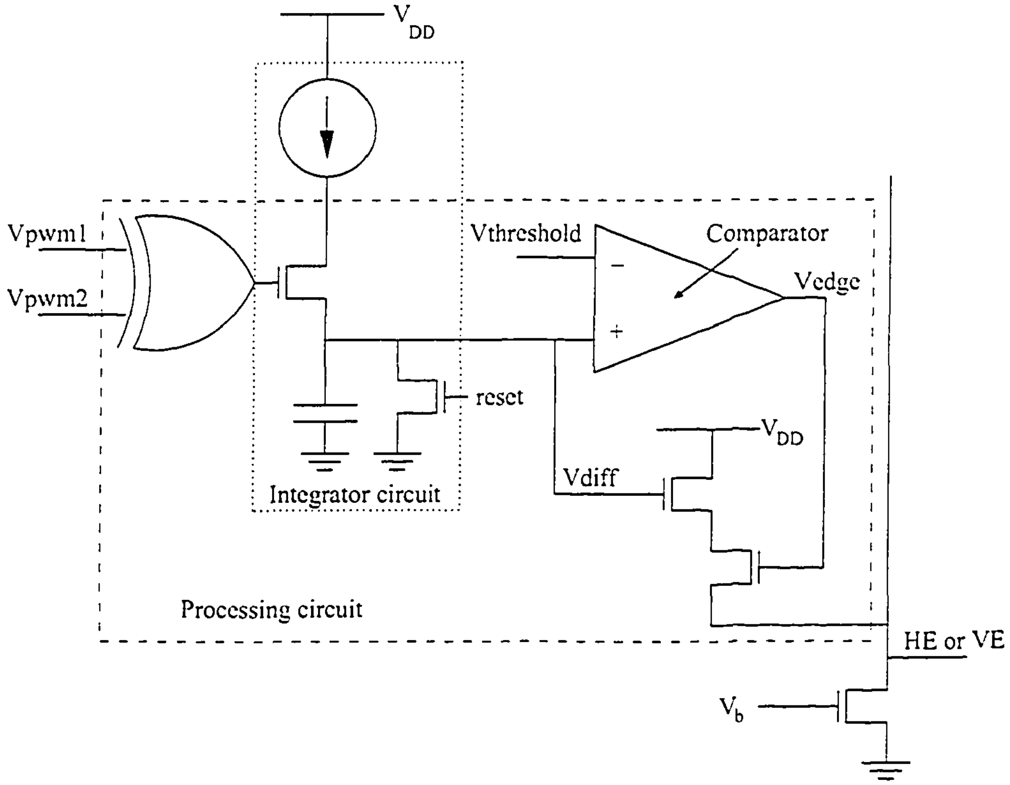


Figure 5.7: Edge detection circuit

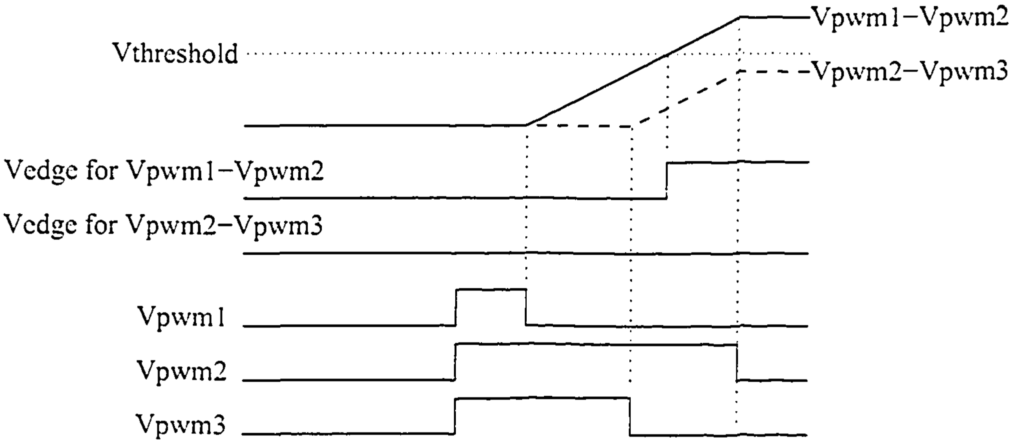


Figure 5.8: Edge detection timing diagram

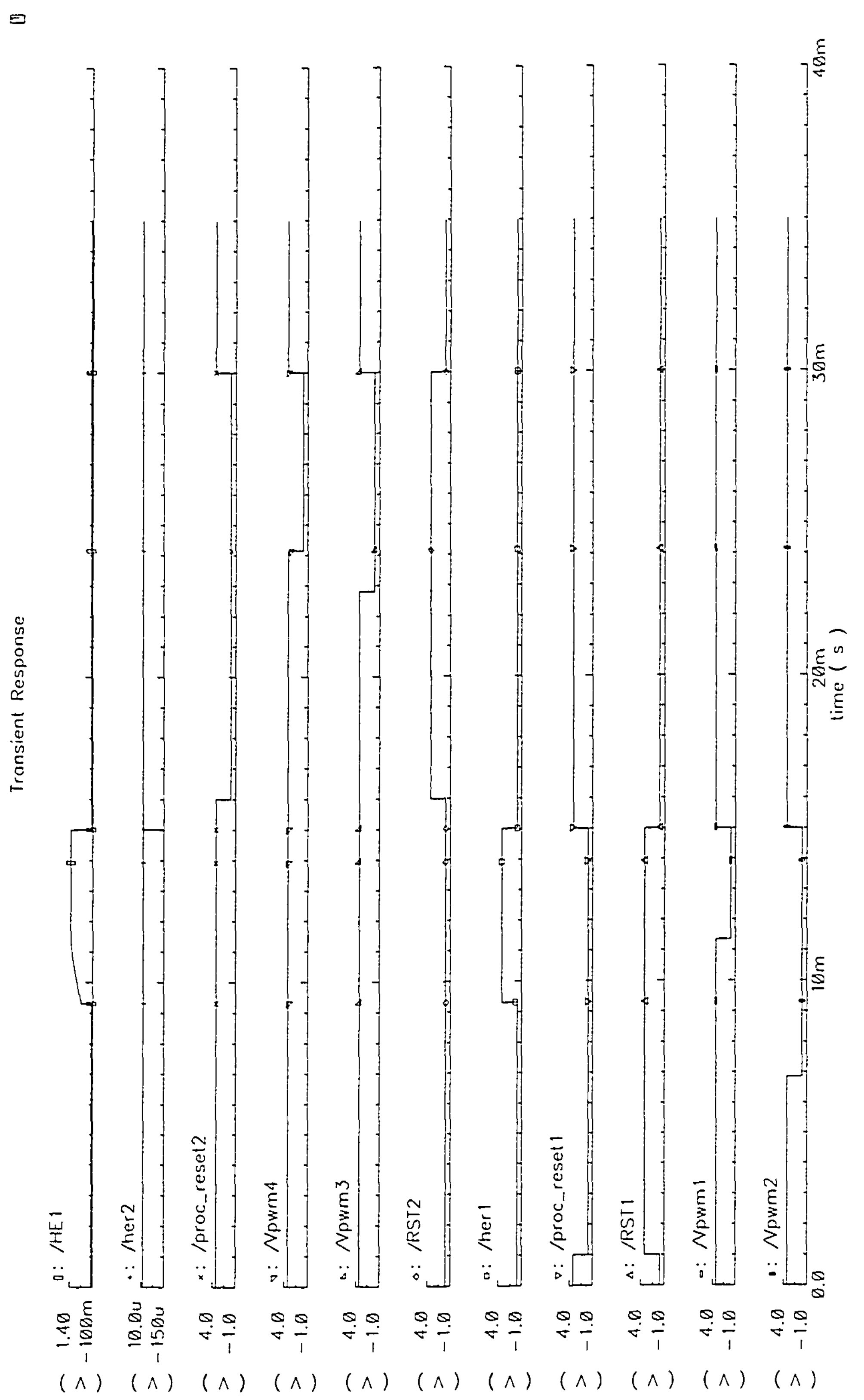


Figure 5.9: Simulation of a 2x2 pixel, row by row readout

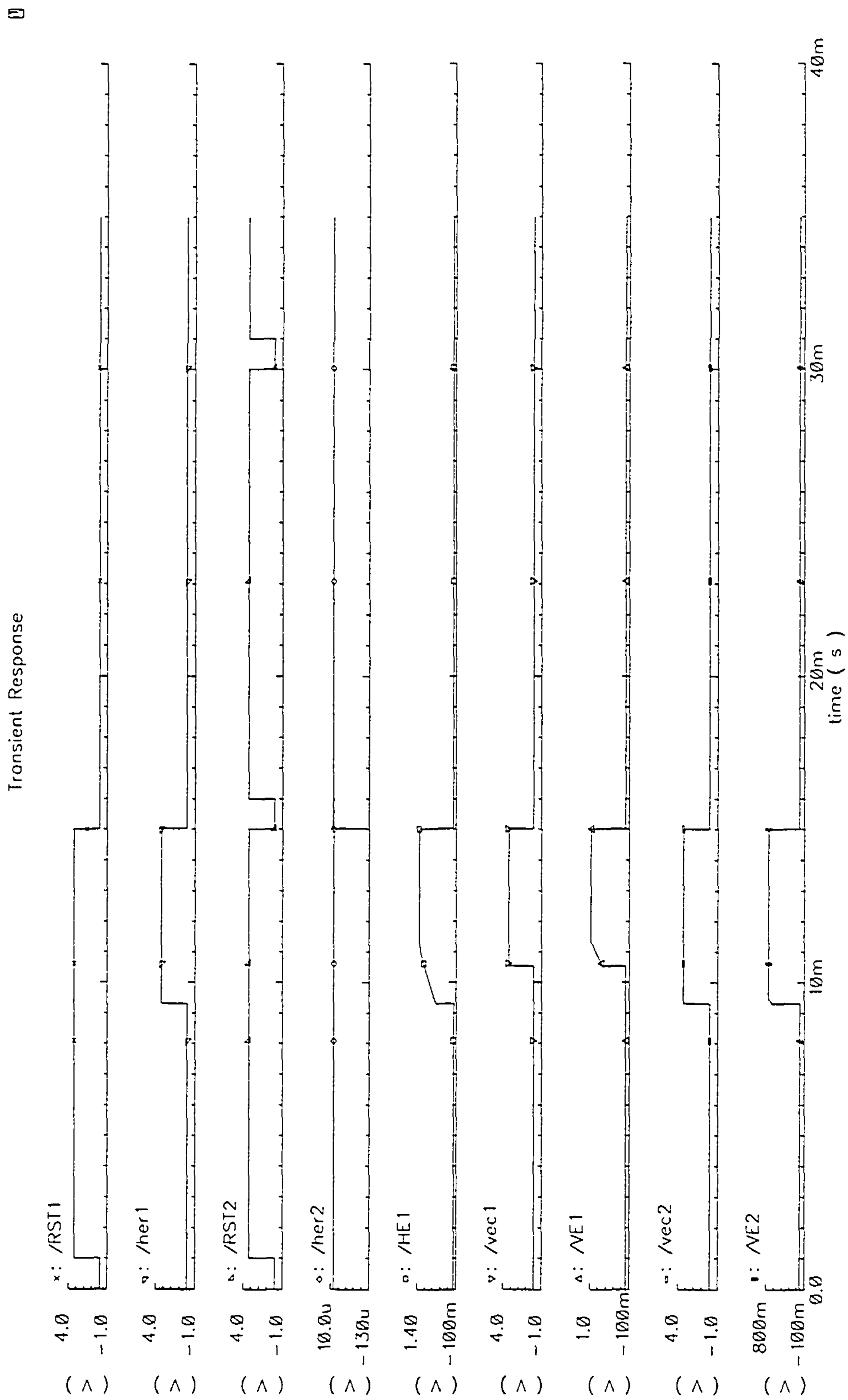


Figure 5.10: Simulation of a 2x2 pixel, 2nd version of readout

source follower and a column line for output. If the flag bit is 0, then V_{diff} is not passed to the column line. There are two methods of readout: one row at a time or two rows at a time. The regular readout is that one row is read out at a time and only the horizontal edges are read out. The two rows at a time readout output the horizontal edges and the vertical edges. From Figure 5.2, column lines like VE1 and HE1 are fed into MUXes for output. Simulations of the proposed pixel are shown in Figures 5.9 and 5.10.

The simulations show two sets of two PWM signals and the output of the edge detection flags, her1, her2, vec1, and vec2. In Figure 5.9, the reset transistor RST1 is off from 1ms to 15ms. V_{pwm1} and V_{pwm2} are XORed and the output of the XOR gate is integrated onto a capacitor just before 7ms. The integrated value is larger than the threshold value at around 9ms and the flag bit her1 is turned on. The actual edge voltages are shown as HE1, VE1 and VE2. For the second cycle, the reset transistor for the second row, RST2, is off from 16ms to 30ms. V_{pwm3} and V_{pwm4} are XORed and the output of the XOR gate is integrated onto a capacitor just before 23ms. The integrated value is not larger than the threshold value and therfore, the flag bit her2 is off. In Figure 5.10, RST1 is off from 1ms to 15ms. Similar to Figure 5.9, her1 is on at around 9ms. The vertical edges are also detected in this readout mode. At around 9ms, vec2 is on to indicate that there is an edge between pixels 2 and 4. For vec1, it is turned on at 10.5ms. For the second cycle, there is no edge between pixels 3 and 4, so her2 is off. The simulation results showed that the proposed pixel and processing circuit can perform edge detection.

5.3 Chapter conclusion

We presented a proposed computational CMOS image sensor based on the idea of incorporating signal processing circuits at the pixel. The use of a simple logic gate to perform edge detection is demonstrated in simulations. The details of the circuit design, simulations and layout are also presented.

Chapter 6

Conclusion

An architecture for a computational CMOS image sensor with built-in edge detection is proposed. Edge detection can be performed by using PWM and single logic gates. One of the advantages of this proposed imager is the simplicity of the analog to PWM circuits. Another advantage is that the PWM signal processing circuits use very few components, mainly the XOR gate and the integrator circuit to perform edge detection. These simple circuits enable us to embed these circuits into the pixel array without taking up a lot of silicon area and to reduce the amount of information for output. Also, by integrating simple image processing functions such as edge detection, system costs and power can be reduced.

This thesis presented a prototype chip with 8 types of pixel designs. The chip was fabricated in a standard 1poly-4metal 0.35μm CMOS process. The test results for the prototype chip were successful and provide insights such as sensitivity on the different pixel designs.

6.1 Future directions

A computational CMOS image sensor probably uses less power than with multi-chip imaging systems at the expense of functionality. This allows imaging systems to be more compact and run longer on batteries, which are important factors for small hand held products. Biometrics is one area in which a computational sensor might be useful. Portable units for fingerprint scanning, retina scanning or facial recognition for law enforcement officers or airport security guards are possible.

Industrial inspection is another application area for these sensors. The sensors can be used to determine different edges on printed circuit boards and other fabricated components. The edges from each board are compared with the edges from a defect-free board and if they are different then there are defects. This can be useful for high speed inspection systems. More sophisticated edge detection than two pixel comparisons are likely to be required.

Another application of computational image sensors is in the medical industry. The medical industry is trying to find ways to reduce pain for patients during procedures. For example, during a colonoscopy, a doctor needs to insert a cable with a camera (a scope) into the patient to check his or her colon. Instead of using a scope, companies have invented a pill, which contains an image sensor. The patient swallows the pill and the pill captures images of the patient's gastrointestinal tract and sends those images through RF. A computational sensor can be used to process the information and send only the useful information, thus increasing efficiency.

The fabrication of the proposed computational image sensor is one of the future directions. Another direction is to embed new algorithms into the image sensor. There is a bright future for computational CMOS image sensors as the feature size of CMOS technology continues to reduce. More signal processing circuits can be added to the pixel without significantly reducing the fill factor. Information can be processed at the pixel level and only the useful information needs to be output.

Bibliography

[1] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, M. Abe, H. Maruyama, and H. Otake. Computational image sensor for on sensor compression. *IEEE Trans. on Electron Devices*, 44(10):1724–1730, 1997.

[2] A. Bermak. A CMOS imager with PFM/PWM based analog-to-digital converter. In *2002 IEEE International Symposium on Circuits and Systems*, volume 4, pages 26–29, 2002.

[3] W. S. Boyle and G. E. Smith. Charge-coupled semiconductor devices. *Bell System Technical Journal*, 49:587–593, 1970.

[4] B. D. Brown and H. C. Card. Stochastic neural computation i: computational elements. *IEEE Trans. on Computers*, 50(9):891–905, 2001.

[5] S. Decker, R. D. McGrath, K. Brehmer, and C. G. Sodini. A 256 x 256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output. *IEEE Journal of Solid State Circuits*, 33(12):2081–2091, 1998.

[6] B. Dierickx, D. Scheffer, G. Meynants, W. Ogiers, and J. Vlummens. Random addressable active pixel image sensors. In *Proc. SPIE*, volume 2950, pages 2–7, 1996.

[7] P. L. P. Dillon, A. T. Brault, J. R. Horak, E. Garcia, T. W. Martin, and W. A. Light. Fabrication and performance of color filter arrays for solid-state imagers. *IEEE Journal of Solid State Circuits*, SC-13(1):23–27, 1978.

[8] K. M. Findlater, D. Renshaw, J. E. D. Hurwitz, R. K. Henderson, M. D. Purcell, S. G. Smith, and T. E. R. Bailey. A CMOS image sensor with a double-junction active pixel. *IEEE Trans. on Electron Devices*, 50(1):32–42, 2003.

[9] E. R. Fossum. CMOS image sensors: electronic camera-on-a-chip. *IEEE Trans. on Electron Devices*, 44(10):1689–1698, 1997.

[10] P. Fry, P. Noble, and R. Rycroft. Fixed pattern noise in photomatrices. *IEEE Journal of Solid State Circuits*, SC-5:250–254, 1970.

[11] B. R. Gaines. Stochastic computing systems. *in J. F. Tou (Ed.): 'Advances in Information System'*, 2:37–172, 1969.

[12] A. El Gamal. *EE392B: Introduction to Image Sensors and Digital Cameras*. Stanford University, Spring 2001.

[13] V. Gaudet and A. Rapley. Iterative decoding using stochastic computation. *Electronics Letters*, 39(3):299–301, 2003.

[14] A. Graupner, J. Schreiter, S. Getzlaff, and R. Schuffny. CMOS image sensor with mixed-signal processor array. *IEEE Journal of Solid State Circuits*, 38(6):948–957, 2003.

[15] T. Hanamoto and K. Aizawa. A computational image sensor with adaptive pixel-based integration time. *IEEE Journal of Solid State Circuits*, 36(4):580 585, 2001.

[16] J. Horton, R. Mazza, and H. Dym. The scanistor - a solid-state image scanner. In *Proc. IEEE*, volume 52, pages 1513–1528, 1964.

[17] T. Kanade and R. Bajcsy. Computational sensors: A report from DARPA workshopr. In *IUS Proc.*, 1993.

[18] S. Kleinfelder, S. Lim, X. Liu, and A. El Gamal. A 10000 frames/s CMOS digital pixel sensor. *IEEE Journal of Solid State Circuits*, 36(12):2049 2059, 2001.

[19] N. Koike, I. Takemoto, K. Sato, H. Matsumaru, M. Ashikawa, and M. Kubo. An NPN structure 484x384 MOS imager for a single-chip color camera. In *IEEE Int. Solid State Circuits Conference*, pages 192–193, 1979.

[20] M. Kyomasu. A new MOS imager using photodiode as current source. *IEEE Journal of Solid State Circuits*, 26(8):1116–1122, 1991.

[21] J. Lamperti. *Probability: a survey of the mathematical theory*. John Wiley & Sons, 1996.

[22] T. Lule, B. Schneider, and M. Bohm. Design and fabrication of a high-dynamic-range image sensor in tfa technoloy. *IEEE Journal of Solid State Circuits*, 34(5):704 711, 1999.

[23] S.-Y. Ma and L.-G. Chen. A single-chip CMOS APS camera with direct frame difference output. *IEEE Journal of Solid State Circuits*, 34(10):1415–1418, 1999.

[24] D. K. McNeill, D. Zhao, C. Shafai, N. Chadha, A. Cuhadar, and H. C. Card. Processing noisy analog signals from microsensor arrays and VLSI imagers using stochastic binary computations. In *Proc. 2002 IEEE Canadian Conference on Electrical and Computer Engineering*, volume 2, pages 986–990, 2002.

[25] R. Melen. The tradeoff in monolithic image sensors: MOS versus CCD. *Electron.* 46:106–111, 1973.

[26] S. Mendis, S. E. Kemeny, and E. R. Fossum. CMOS active pixel image sensor. *IEEE Trans. on Electron Devices*, 41(3):452–453, 1994.

[27] R. Merrill. Color separation in an active pixel cell imaging array using a triple-well structure. *U.S. Patent 5965875*, 1999.

[28] S. Mizuno, K. Fujita, H. Yamamoto, N. Mukozaka, and H. Toyoda. A 256 x 256 compact CMOS image sensor with on-chip motion detection function. *IEEE Journal of Solid State Circuits*, 38(6):1072–1075, 2003.

[29] Alireza Moini. Vision chips or seeing silicon. Technical report, The University of Adelaide, 1997.

[30] M. Mori, M. Katsuno, S. Kasuga, T. Murata, and T. Yamaguchi. A 1/4in 2m pixel CMOS image sensor with 1.75transistor/pixel. In *IEEE Int. Solid State Circuits Conference*, pages 110–111, 2004.

[31] S. Morrison. A new type of photosensitive junction device. *Solid-State Electron.* 5:485–494, 1963.

[32] Y. Muramatsu, S. Kurosawa, M. Furumiya, H. Ohkubo, and Y. Nakashiba. A signal-processing CMOS image sensor using a simple analog operation. *IEEE Journal of Solid State Circuits*, 38(1):101–106, 2003.

[33] M. Nagata, M. Homma, N. Takeda, T. Morie, and A. Iwata. A smart CMOS imager with pixel level PWM signal processing. In *Digest of Technical Papers 1999 Symposium on VLSI Circuits*, pages 141–144, 1999.

[34] P. Noble. Self-scanned silicon image detector arrays. *IEEE Trans. on Electron Devices*, ED-15:202–209, 1968.

[35] M. Ogata, T. Nakamura, K. Matsumoto, R. Ohta, and R. Hyuga. A small pixel cmd image sensor. *IEEE Trans. on Electron Devices*, 38:1005–1010, 1991.

[36] S. Ohba, M. Nakai, H. Ando, S. Hanamura, S. Shimada, K. Satoh, K. Takahashi, M. Kubo, and T. Fujita. MOS area sensor: Part ii - low noise MOS area sensor with anti-blooming photodiodes. *IEEE Trans. on Electron Devices*, ED-27:1682–1687, 1980.

[37] J. Ohta, K. Yamamoto, T. Hirai, K. Kagawa, M. Nunoshita, M. Yamada, Y. Yamasaki, S. Sugishita, and K. Watanabe. An image sensor with an in-pixel modulation function for detecting the intensity of a modulated light signal. *IEEE Trans. on Electron Devices*, 50(1):166–172, 2003.

[38] K. A. Parulski. Color filters and processing alternatives for one-chip cameras. *IEEE Trans. on Electron Devices*, ED-32(8):1381–1389, 1985.

[39] R. J. Perry. Analysis and characterization of the sprctral response of CMOS based integrated circuit (IC) photodetectors. In *Proc. the Thirteenth Biennial, University, Government, Industry Microelectronics Symposium*, pages 170–175, 1999.

[40] D. Renshaw, P. Denyer, G. Wang, and M. Lu. ASIC image sensors. In *1990 IEEE International Symposium on Circuits and Systems*, pages 3038–3041, 1990.

[41] K. Senda, S. Terakawa, Y. Hiroshima, and T. Kunii. Analysis of charge-priming transfer efficiency in CPD image sensors. *IEEE Trans. on Electron Devices*, ED-31:1324–1328, 1984.

[42] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi, and G.-F. Dalla Betta. Novel CMOS image sensor with a 132-dB dynamic range. *IEEE Journal of Solid State Circuits*, 37(12):1846–1852, 2002.

[43] S. M. Sze. *Physics of semiconductor devices*. John Wiley & Sons, 1981.

[44] H. Takahashi, M. Kinoshita, K. Morita, T. Shirai, T. Sato, T. Kimura, H. Yuzurihara, and S. Inoue. A 3.9μm pixel pitch VGA format 10b digital image sensor with 1.5-transistor/pixel. In *IEEE Int. Solid State Circuits Conference*, pages 108–109, 2004.

[45] N. Tanaka, S. Hashimoto, M. Shinohara, S. Sugawa, M. Morishita, S. Matsumoto, Y. Nakamura, and T. Ohmi. A 310 k-pixel bipolar imager (BASIS). *IEEE Trans. on Electron Devices*, 38:964–971, 1990.

[46] H. Tian, B. Fowler, and A. El Gamal. Analysis of temporal noise in CMOS photodiode active pixel sensor. *IEEE Journal of Solid State Circuits*, 36(1):92–101, 2001.

[47] C.-C. Wang and C. G. Sodini. The effect of hot carriers on the operation of CMOS active pixel sensors. In *Proc. IEEE Electron Devices Meeting*, pages 24.5.1–24.5.4, 2001.

[48] G. P. Weckler. Operation of p-n junction photodetectors in a photon flux integration mode. *IEEE Journal of Solid State Circuits*, SC-2:65–73, 1967.

[49] C. Xu, C. Shen, A. Bermak, and M. Chan. A new digital-pixel architecture for CMOS image sensor with pixel-level ADC and pulse width modulation using a 0.18 μm CMOS technology. In *2003 IEEE Conference on Electron Devices and Solid-State Circuits*, pages 265–268, 2003.

[50] C. Xu, W. Zhang, W.-H. Ki, and M. Chan. A 1.0-V VDD CMOS active-pixel sensor with complementary pixel architecture and pulsewidth modulation fabricated with a 0.25 μm CMOS process. *IEEE Journal of Solid State Circuits*, 37(12):1853–1859, 2002.

[51] O. Yadid-Pecht and A. Belenky. In-pixel autoexposure CMOS APS. *IEEE Journal of Solid State Circuits*, 38(8):1425–1428, 2003.

[52] D. X. D. Yang, A. El Gamal, B. Fowler, and H. Tian. A 640 x 512 CMOS image sensor with ultrawide dynamic range floating-point pixel-level ADC. *IEEE Journal of Solid State Circuits*, 34(12):1821–1834, 1999.

[53] K. Yoon, C. Kim, B. Lee, and D. Lee. Single-chip CMOS image sensor for mobile applications. *IEEE Journal of Solid State Circuits*, 37(12):1839–1845, 2002.

[54] W. Zhang and M. Chan. Properties and design optimization of photo-diodes available in a current CMOS technology. In *Proc. IEEE Hong Kong Electron Devices Meeting*, pages 22–25, 1998.