Technical University of Crete

School of Electronics and Computer Engineering

Development of a Smart and Reconfigurable Electronic Imaging System

Andreas Symeou

Thesis Committee

Balas Costas, Professor, Supervisor

Kalaitzakis Konstantinos, Professor

Pnevmatikatos Dionisios, Professor

October, 2015

ABSTRACT

The continuous development of hardware performance, in order to achieve higher data bandwidth transfer and integrate peripherals with minimal time and effort, has led to consumer available products capable of more specialized applications. For many years the development of an advanced, smart and reconfigurable electronic imaging system was extremely difficult due to hardware limitations. Modern solid state imagers deliver high image quality with high frame rates, when on the same time, driven by the industry trend for better performing cameras, have promising future development. On the other hand, high performance from the supporting hardware is demanded. This thesis introduces an innovative reconfigurable platform supporting various imagers with different characteristics and capable of controlling imaging parameters. The platform exploits characteristics of the USB Video Class (UVC) to transfer the RAW pixel data captured by the imager, achieving high frame rates. Image quality varies as the platform can exploit modern hardware to process the frames captured by the imager, according to the demands of a given application. As a first step the system is introduced in applications such as low light imaging, real time human vision emulation and image processing with high image quality.

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisor professor Costas Balas for his trust and guidance during the course of this thesis. Also special thanks to professor Pnevmaticatos Dionisios for the advice he provided with the ongoing research and to professor Kalaitzakis Konstantinos.

Next, I would like to thank the members of the Electronics Laboratory for the valuable help they provided, namely Christos Rwssos for the help he provided building the PCB and supporting me through the development of this thesis, next I would like to thank Theodoros Marios Giakoumakis for his great assistant with the application code. I also like to thank Alexis Ntousakis for the help he provided for the system performance evaluation. Special gratitude must be addressed to Ph.D candidate Thanasis Tsapras; for without his guidance, technical expertise and help this thesis would not have been completed.

Next, I would like to thank all the members of the Electronics Laboratory who create a fun working environment, shared ideas with me and supported me.

Last but not least I would like to thank my family and friends for all the help and support through all these years.

Table of Contents

1 INTRODUCTION 9

1.1 Motivation for the design 9

1.2 Objective 10

1.3 Approach 10

1.4 Thesis Overview 11

2 THEORETICAL BACKGROUD 12

2.1 Key Components of an electronic imaging system 12

2.2 Optical system 13

2.3 Solid State Digital Image Sensors 15

2.3.1 CCD image Sensors 17

2.3.2 CMOS image Sensors 21

2.3.3 Imagers Integration technology 24

2.3.4 CCD and CMOS image Sensor Comparison 25

2.4 Frame Acquisition 28

2.5 Image Processing 31

2.6 Video Representation 37

2.6.1 YUV Color space 37

2.6.2 RGB Color space 38

2.7 Imaging System Performance 39

2.7.1 Modulation Transfer Function – MTF 40

2.7.2 Resolution Test 44

3 RELATED WORK 45

3.1 Objective Oriented 45

3.2 Hardware Oriented 46

4 PROPOSED SOLUTIONS 47

4.1 Multicore Embedded system-on-Chip – SoC 47

4.2 General Purpose Processors 48

4.3 Field Programmable Gate Arrays – FPGAs 49

4.4 Data Acquisition – DAQ Cards 50

4.5 Serial Interface Frame Grabbers 52
4.6 Embedded Video Interfaces 53
4.7 Conclusion 54
5 PLATFORM IMPLEMENTATION 55
5.1 EZ-USB FX3 Controller Architecture 55
5.1.1 GPIF – II Peripheral 57
5.1.2 $I2C$ Interface 59
5.1.3 JTAG Interface 59
5.1.4 UART Interface and Integrated Debugger 60
5.2 Hardware Implementation 61
5.2.1 MT9M034 CMOS Image Sensor 61
5.2.2 Image Sensor Board 64
5.2.3 Image Sensor Board Optics 66
5.3 FX3 Firmware 66
5.3.1 USB Video Class – UVC 68
5.3.2 Video Data Format: YUY2 and RGB24 69
5.3.3 GPIF II State Machine 70
5.3.4 FX 3 DMA System 71
6 PERFORMANCE EVALUATION 78
6.1 Demosaicing 78
6.2 System Spatial Resolution 81
6.2.1 Modulation Transfer Function – MTF Measurement 81
7 DEVELOPED APPLICATIONS 84
7.1 Imaging at low light conditions 84
7.2 Real Time Human Vision Emulation 87
8 CONCLUSIONS 91
8.1 Sensor Connectivity 91
8.2 FX3 Controller and firmware development 91
8.3 Overview of the system performance 92
9 FUTURE WORK 93
9.1 This design 93
9.2 A Platform Design 93
10 References 94

Figure 2.3.1-1 Imaging pipeline of a modern electronic imaging system.....12
Figure 2.3.1-2 The four main modules of the system. 13
Figure 2.3.1-1 Diffraction pattern of red laser beam made on a plate after passing a small circular hole in another plate 14
Figure 2.3.1-2 Examples of chromatic aberration and correcting methods. 15
Figure 2.3.1-1 MOS Capacitors and the reaction from photoelectric effect. 16
Figure 2.3.1-2: n-MOS Linear image sensor from Hamamatsu.....16
Figure 2.3.1-3 Schematic diagram of the correlated double sampling circuit (Left) and a) Active Pixel Array – APS photocircuit using a photogate. b) Using a photodiode (Right).....17
Figure 2.3.1-1 The three CCD architectures: a) Full frame, b) Frame-transfer and c) Interline-transfer ...19
Figure 2.3.1-2 a) Normal image, b) image with blooming, c) image with blooming and saturation when maximum charge transfer capacity of the CCD is reached. 20
Figure 2.3.1-3 LOD allows electron overflow directly into the substrate rather than moving laterally into a drain channel.....21
Figure 2.3.2-1 The three main CMOS architectures: a) Passive Pixel Sensor – PPS, b) Active Pixel Sensor – APS, c) Digital Pixel Sensor – DPS.....23
Figure 2.3.2-2 The PPD integrated to an APS structure 23
Figure 2.3.3-1 The sCMOS developed by Imec 24
Figure 2.3.3-2 FSI pixel on the left and the new BSI pixel in the Right 25
Figure 2.3.4-1 Integration potential of the CCD technology and CMOS technology.....25
Figure 2.3.4-1 The Frame Valid – FV and Line Valid – LV signals indicating the valid frame pixel readouts. The frame stars after the first logic true in FV&LV (first A). After the desertion of LV the first line of the frame has arrived. The process is repeated until.....29
Figure 2.3.4-2 In more detail this figure shows the data readout from a sensor with 12 bit parallel interface. The valid pixel values are indicated by the Pixel Clock – PIXCLK.....29
Figure 2.3.4-3 The LVDS Data lane with a driver and a terminal receiver. 30
Figure 2.3.4-4 The Bridge converting a serial interface to parallel above and the IP core enabling deserialization of 4 lanes of serial data below. 31
Figure 2.3.4-1 Above the beam splitter color separation with A, B and C CCD detectors and below a Bayern CFA 32
Figure 2.3.4-2 Some Demosaicing artifacts. 33
Figure 2.3.4-3 : The Aliasing artifact (above) presented when there is aliasing between the green, red and blue spectrums. The zipper effect is presented when pixels of a uniform area in image get mixed together by the demosaicing algorithm with another uniform area 34
Figure 2.3.4-4 The four cases of pixel interpolation in NN algorithm.....34
Figure 2.3.4-5 Bilinear interpolation with a 4 X 4 window 36
Figure 2.6.2-1 The sRGB color triangle, shown as a subset of x,y space, a chromaticity space based on CIE 1931 colorimetry 39
Figure 2.7.1-1 Imaging Resolution Scenarios where (a) the line-pair is not resolved and (b) the line-pair is resolved [35 – page 74].....41
Figure 2.7.1-2 The MTF of a system at a low and high frequency and the MTF curve 42

Figure 2.7.1-3 Contrast expressed as a square wave at different levels of resolution. 43
Figure 2.7.2-1 Grill Pattern Target. 44
Figure 2.7.2-1 The OMAP-L138 Low Cost Development Kit (OMAP-L138 LCDK) 45
Figure 2.7.2-1 The PCB created with the FX2LP microcontroller. 46
Figure 2.7.2-1 The Exynos 5-based Arndale community board for application developers. 47
Figure 2.7.2-1 The Intel ATOM CPU interconnection with 3 different imaging sensors. 48
Figure 2.7.2-1 Xilinx Vertex 6 FPGA with a PCI-X communication for fast image data transmission. 49
Figure 2.7.2-1 The Frame grabber featuring a DAQ card and a bridge to desirialize the data and make them readable to the card. 50
Figure 2.7.2-1 The complete frame grabber solution. 52
Figure 2.7.2-1 The three boards found featuring the FX3 USB3 controller: On the top left the CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress, next the Net Vision SVI-06 and below the Lattice USB 3 Board , all offering great platform reconfigurability and hardware performance. 54
Figure 2.7.2-1 The functional diagram of the EZ-USB FX3 controller. 55
Figure 2.7.2-2 ARM926EJ-S Block Diagram 56
Figure 5.1.1-1 The GPIF II Designer Tool enables the interface parameter selection resulting in an easy and fast implementation of the interface data with, communication blocks utilized and specific control characteristics. 58
Figure 5.1.4-1 Block Diagram of CY7C65215. 60
Figure 5.2.1-1 Block Diagram of MT9M034 CMOS image sensor. 61
Figure 5.2.1-2 Color filter pattern (Bayern pattern) of sensor's active pixel array. 62
Figure 5.2.1-3 Spatial representation of image readout. 63
Figure 5.2.1-4 Front view of the LI-CAM-M034 camera board with a 6mm lens mounted. 64
Figure 5.2.2-2 The molex FFC connector used for the PCB fabrication. 64
Figure 5.2.2-3 The necessary connection between the controller board and camera board. 65
Figure 5.2.2-4 The final hardware connected and fully functional. 65
Figure 5.2.3-1 The firmware stack on the left and the Firmware Sequence on the right. 67
Figure 5.3.1-1 UVC Diagram of the Camera Architecture 68
Figure 5.3.2-1 Illustration for the UVC Video Data Transfers 69
Figure 5.3.3-1 The Final State Machine Diagram Designed with GPIF – II Designer software. 70
Figure 5.3.4-1 GPIF II Socket/Thread Mapping 71
Figure 5.3.4-2 A Socket Routes Data According to a List of DMA Descriptors. 72
Figure 5.3.4-3 A Socket Operating with DMA Descriptor 1 72
Figure 5.3.4-4 A Socket Operating with DMA Descriptor 2. 73
Figure 5.3.4-5 A Socket Operating with DMA Descriptor 3. 73
Figure 5.3.4-6 DMA Transfer Example. 74
Figure 5.3.4-7 Image Sensor Interface, Data Path Execution, and State Machine Correlation. 76
Figure 5.3.4-8 Dual Sockets Yield Seamless Transfers. 76
Figure 6.1.1-1 Contour Stencils demosaicing done offline. The image is taken from the book hardcover: Digital Image Processing (3rd Edition) [6.1]. 79
Figure 6.1.1-2 A detail taken from the first image indicating the differences between the various stages of the frame processing and a closer look to the performance of the bilinear interpolation demosaicing. First at the up left corner is the RAW pixel values taken from the imager. The second image on the up right corner presents the "colored" RAW pixel intensity values according to the Bayern CFA. At the

bottom the bilinear interpolation demosaicing (left) and the Contour Stencils algorithm (right) are presented. 80
Figure 6.2.1-1 Pattern after Bilinear interpolation demosaicing 82
Figure 6.2.1-2 Pattern after Contour Stencils demosaicing algorithm with alpha parameter equal to 1.2.. 82
Figure 6.2.1-3 Pattern after Contour Stencils with alpha parameter set to 2. 82
Figure 6.2.1-4 Pattern produced by the RAW pixel intensities without any demosaicing process. 82
Figure 7.1.1-1 : The result of a long exposure image captured by the CMOS sensor. 85
Figure 7.1.1-2 The result of 2 averaged “dark frames” 85
Figure 7.1.1-3 Final result of the de-noise procedure with dark frame. 86
Figure 7.2.1-1 The “pixels” of the human eye. 87
Figure 7.2.1-2 The spatial distribution of cones and robs. 88
Figure 7.2.1-3 Detail of the received frame from the human vision application. 89
Figure 7.2.1-4 FPN presented by high frame rates. 90

1 INTRODUCTION

1.1 Motivation for the design

For many years the development of an advanced, smart and reconfigurable electronic imaging system was extremely difficult due to hardware limitations such as memory, bandwidth and processing capabilities. The continuous development of hardware the past few years to achieve higher data bandwidth transfers and interface peripherals with minimal integration time and effort has led to consumer available hardware capable of such task.

One of the most important parts of the system is the Digital image sensor. Modern imagers have made great advancements over the last years driven mainly by the smartphone industry to make high performance sensors capable of DSLR image quality[1]. Unfortunately the image resolution that smartphones can reach is significantly restricted by physical factors like the optical system resolution (more at Chapter 2.1.1). Modern imaging systems have the capability to communicate with image sensors via many physical layers in order to give smart features to the final user[2]. A disadvantage of this approach is that the platform communication with the image sensor tends to be complicated, and final user control over the imager to be extremely difficult and restricted. Another drawback is the limited number of image sensors supported by this kind of platforms.

The majority of existing imaging systems give access to ether compress data (ex. MJPG) or uncompressed RGB data and lack the ability to choose the image processing algorithm or to exploit the RAW pixel readings for offline processing (higher image quality) or metrics extraction (Machine Vision). Knowing the exact pixel values from the digital image sensor is essential for many applications and can boost the frame rate of the system depending on the Host performance (ex. PC).

Today's Personal Computers have powerful Graphics Processing Units – GPUs and Central Processing Units – CPUs capable of various image and video processing tasks in real time. A modern PC can be used for processing the image data captured directly from a Digital solid state imager using ether a CPU or GPU for higher performance due to parallelization of the image processing workload. The main advantage of this approach is the remaining performance of the platform that can be used for transferring higher bandwidth data to the final application.

---

[1] http://evan-theelectricalengineer.blogspot.gr/2013/10/nikon-d800e-vs-nokia-lumia-1020.html

[2] https://developer.android.com/reference/android/hardware/camera2/package-summary.html

The future development of the system is very promising with new hardware available and new sensors with different characteristics. The peripheral controller selected can be used by many inexpensive or high-end platforms, so with some changes the new hardware can be integrated into the system. On the imagers side the control over some of the sensors characteristics like the Region Of Interest – ROI, a feature mainly useless to smartphone cameras, can be exploit by such platform.

1.2 Objective

The goal of this thesis is to successfully design and develop a platform which can take advantage of the image digital sensor characteristics and abilities, transfer high bit rates to a host application, develop on-the-fly sensor access and give an acceptable frame rate (> 30 FPS - Frames Per Second) and image quality (at least 1080 x 720). Moreover the platform must be able to interface with different image sensors with minor changes to hardware and software and change the image processing algorithm to meet the image quality vs performance demands.

Finally the imaging system will be tested for image quality and review the performance of some applications like Human Vision emulation, low light imaging and De-noising with frame averaging.

1.3 Approach

In order to make a reconfigurable imaging system it's essential to discuss the capability of the platform to interface with various imagers and create a smart scientific grade electronic imaging system featuring state of the art sensors. High-end sensors have high bandwidth data transfer capabilities to achieve more frame rate at higher image resolutions. The need for the platform to process and transfer data at such speeds has high performance demands. If the platform cannot compensate frame rate and image quality at acceptable levels then some image sensors with high end characteristics may be unusable.

To avoid that the peripheral of the platform will focus only on transferring the pixel data from the digital image sensor and leave the Image Signal Processing – ISP to the host application. This is an unusual approach for an imaging system due to the hardware efficiency of most embedded systems to process image data.

The benefits of this approach are mainly the reconfigurability of the system to use different algorithms for image processing and give the platform hardware more processing power for high bandwidth data transfers and dealing with the demanding interfaces of modern imagers (deserialize frame data, etc.). Moreover the hardware of a PC can be easily upgraded or a more

powerful updated PC can be used with the same application running the same or new algorithms more efficiently than the old one. To make this possible the system will need standard generic drivers used from all modern PCs in order to function in any new PC without any problems.

The peripheral connecting the PC and platform must be fast enough to transfer at least 720p@30 fps which amounts to nearly 560Mbps using RGB8 with some overhead for protocol error codes or Cyclic Redundancy Codes - CRC. Strictly speaking the connection bandwidth required for transmitting the RAW pixel data from the sensor to the host application will be: $(1080 * 720) frame\ size * 8\ bits\ per\ pixel * 30\ frames\ per\ second = 186.6 Mbps.$ The bitrate for RGB24 data for this frame size and rate will be 3 times more than the actual pixel values from the image sensor. Connections capable of providing the necessary bandwidth and bidirectional communication for control are Gig Ethernet, USB and PCI-X which will be discussed in Chapter 2 in more detail.

1.4 Thesis Overview

The thesis outline consist of the main chapters where the different aspects examined by this thesis are explained, and is further divided into topics to simplify the structure and help the reader search thru the text with ease.

In Chapter 2 the most important information consisting the theoretical background is presented with frequent references to the rest of the text. Chapter 3 reviews related work done in the past and the results from this work. Chapter 4 is a detailed reference to the research done for this design. The platform implementation is described in detail by Chapter 5. Then the results and system performance is evaluated by Chapter 6. The applications developed by this design are examined by Chapter 7. Chapters 8 and 9 are the conclusions and future work for providing an overview of the design and help understand the potentials in a future development.

2 THEORETICAL BACKGROUD

2.1 Key Components of an electronic imaging system

The design of an electronic imaging system is a complex task and its components can take very specialized form from system to system. For example in most imaging systems the image processing is done at the start of the image acquisition with advanced hardware pipeline and in most cases 3A algorithms (Auto Exposure - AE, Auto White Balance - AWB, Auto Focus – AF). Below in figure 2.1-1 an example is presented where automatic adjustments to camera lens, tone mapping and exposure are controlled by the system[3].

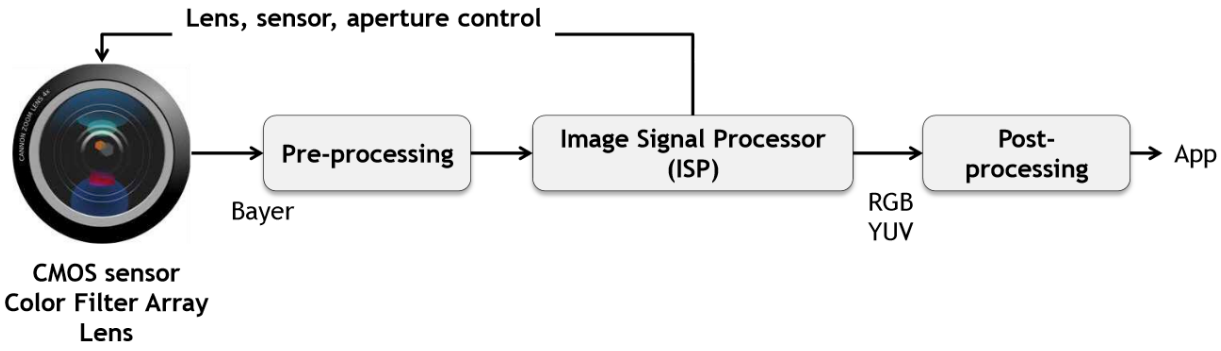


Figure 2.3.1-1 Imaging pipeline of a modern electronic imaging system

In order to simplify and take control of some functions we generalize some key modules divided by the functionality and connectivity they provide as shown in figure 2.1-2 below

[3] https://www.khronos.org/assets/uploads/developers/library/2013-siggraph-camera-bof/Camera-BOF_SIGGRAPH-2013.pdf

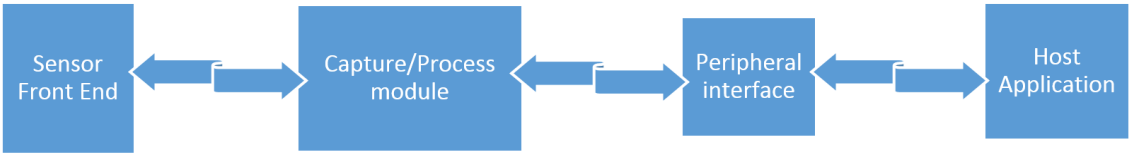


Figure 2.3.1-2 The four main modules of the system.

The first module is the Cmos digital image sensor with the front end electronics essential for the functionality of the sensor chip. The second module is responsible for the acquisition of frame or data with some control signals indicating the start and the end of a frame. Alternatively some image signal processor algorithms can take place in this module hardware without compromising the frame rate performance of the system. The peripheral interface module connects the capture/process module and host application with a fast enough protocol to transfer the images at acceptable frame rate. The host application is the final stage where the frames are processed, represented and/or stored. Processing frames at the host application can exploit available hardware like a GPU and boost the system performance. The optical system is not present to the key components because it's irrelevant for this design functions (no 3A algorithms and control over mechanical iris) and can be change without any modifications to the platform. Details about the optical system can be found to the next topic, explaining how an optical system works and how it affects the rest of the system.

Although the image processing is mandatory for an electronic imaging system, some of the image enhancement algorithms and color reconstruction (demosaicing) can easily run in real time by a modern pc. An image can be formed by intensity distribution as a function of spatial position. This information is enough to reconstruct a perfect color image if the exposure and optics is right and the color filter array – CFA pattern known (more at Chapter 2.1.4). The real time process of images on a pc is possible thru multimedia extensions to microprocessors and the advancements graphics processing units – GPU have made the past few years, able to process teraflops of data.

More specialized figures containing functional details can take place as we review more specific solutions at Chapter 4.

2.2 Optical system

Besides some lensless applications[4] [2.34] optics are essential for the proper function of an imaging system. The light needs to be collected and projected to a photosensitive surface in order to be

[4] http://www.technologyreview.com/view/515651/bell-labs-invents-lensless-camera/
https://www.youtube.com/watch?v=33OqhRHa4pU

captured by the system. The simplest optical system is the pinhole lens[5]. A more improved version are lenses capable to collect more light and focus a light beam through refraction to the photosensitive surface. The optical system is the first limiting factor to a modern imaging system resolution [2.1]. The smallest quantifiable information for a digital image system is the pixel. Pixels tend to be small (< 2μm) which can result to good spatial resolution but poor dynamic range (smaller pixel well). Optics on the other hand are significantly limited by the lens diameter. This is the result of light diffraction when the light collected by the lens acts like a wave and creates constructive and destructive interference. In classical physics, the diffraction phenomenon is described as the interference of waves according to the Huygens–Fresnel principle [2.2]. The result of this phenomenon is the Point Spread Function – PSF which occurs when a point source of light is spreaded to s surface. When observed the interference phenomenon creates the Airy diffraction disk when constructive and destructive interferences occur as shown in figure 2.2-1. When 2 of these sources spreaded by an angle small enough that their Airy disks on the camera detector start overlapping then the sources or objects cannot be clearly separated any more in the image, and they start blurring together. According to Airy function the limit of the smallest angular separation two objects can have before they significantly blur together is given by $sin\theta = 1{,}22\frac{\lambda}{D}$ where θ is the angular separation of the two objects, λ is the light wavelength and D is the lens diameter (aperture). Thus, the ability of the system to resolve detail is limited by the ratio of $\frac{\lambda}{D}$. The larger the aperture for a given wavelength, the finer the detail which can be distinguished in the image. For small angular separation the above function can be simplified as $\theta = 1{,}22\frac{\lambda}{D}$ where θ the angular separation of the two objects in radians. This limitation causes image resolution degradation and affects the overall system image resolution.

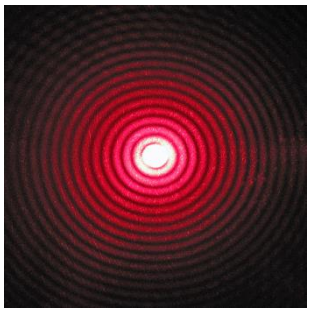
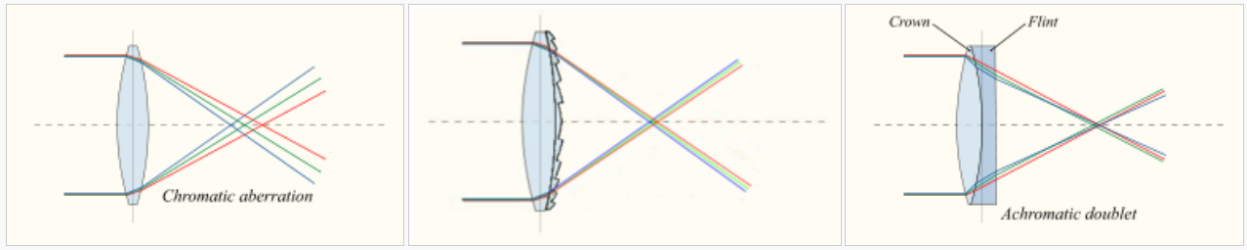


Figure 2.3.1-1 Diffraction pattern of red laser beam made on a plate after passing a small circular hole in another plate

Another limitation for optical systems is the chromatic aberration observed when light of different wavelengths changes focal distance. This is caused by the refractive index of transparent materials that decreases with increasing wavelength in degrees. The solution to this deficiency are apochromatic lenses with more elements or coating filters in order to change the focal distance of different wavelengths to the same point or small area as shown in figure 2.2-2.

[5] https://en.wikipedia.org/wiki/Camera_lens



Chromatic aberration of a single lens causes different wavelengths of light to have differing focal lengths

Diffractive optical element with complementary dispersion properties to that of glass can be used to correct for color aberration

For an achromatic doublet, visible wavelengths have approximately the same focal length

Figure 2.3.1-2 Examples of chromatic aberration and correcting methods. [6]

Image resolution deficiency is more common to modern imaging systems than chromatic aberration and more difficult to compensate and in some cases the only limiting factor to a system (ex. Smartphone cameras with small lens aperture).

The optics of the system developed in the scope of this thesis uses the same optics thru any performance test and the reference to the optical system limitations are for further understanding the impact of optics to the image resolution tests (Chapter 6).

2.3 Solid State Digital Image Sensors

The incident light collected by the optical system needs a photosensitive surface to be converted to electronic charge due to the photoelectric effect and then to be amplified and converted to digital signals. All photonic imagers operate with the photoelectric effect as operating principle. When a surface is exposed to electromagnetic radiation above a certain threshold frequency determined by the material energy gap (typically visible light for alkali metals - Eg1, near ultraviolet for other metals - Eg2, and extreme ultraviolet or x-ray for non-metals - Eg3), the radiation is absorbed and electrons are emitted. The materials energy gap are the main reason why some photonic imagers have better quantum efficiency – QE in certain wavelengths. For example the Metal Oxide Semiconductor - MOS capacitors for each material described above must have the proper energy gaps in the crystalized silicon to be more sensitive to the right wavelength starting from the energy gap with the smallest difference between valence band and conduction band to the bigger for detecting photons with more energy (Eg1 < Eg2 < Eg3) [2.1].

The MOS Capacitor collects the electric charge after the conversion by the photoelectric effect and can be a p – type or n – type. The different types are created with heavily doped silicon and create free electrons for the n –type and free holes for the p – type as shown in figure 2.3-1 [2.3]. Then by incident radiation at p – type the electrons created move to occupy holes to the depletion layer or “potential well” and in n – type they move away from the depletion layer as shown in figure 2.3-1.

[6] https://en.wikipedia.org/wiki/Chromatic_aberration

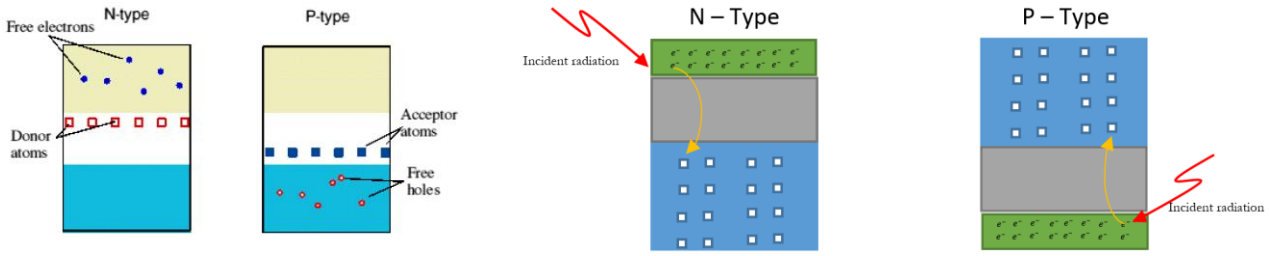


Figure 2.3.1-1 MOS Capacitors and the reaction from photoelectric effect.

Most imagers use p – type MOS capacitors because the speed of electrons is greater than holes in n – type caps. On the other hand n – type imagers have greater well capacity, suitable for spectrometers (linear sensors featuring only a line of pixels, fast readable) giving more dynamic range (figure 2.3-2).

Figure 2.3.1-2: n-MOS Linear image sensor from Hamamatsu

The final pixel architecture is divided in Photodiode and Photogate [2.4]. Photodiode is just a pn – junction creating a depletion layer like a normal diode but with electrons freed by incident radiation changing the depletion layer proportional to the amount of light being absorbed by the pixel. Photogate is a voltage induced junction externally biased thru polysilicon gate to induce a depletion layer in the silicon. The depletion layer changes the same way with incident radiation like photodiode. The photogate designs have worst quantum efficiency (about two times less than the photodiode), especially in the short-wavelength (blue) region of the spectrum. Photogate devices usually have larger pixel areas, but a lower fill factor and much poorer blue light response (and general quantum efficiency) than photodiodes. However, photogates often reach higher charge-to-voltage conversion gain levels and can easily be utilized to perform correlated double sampling to achieve frame differencing (figure 2.3-3).

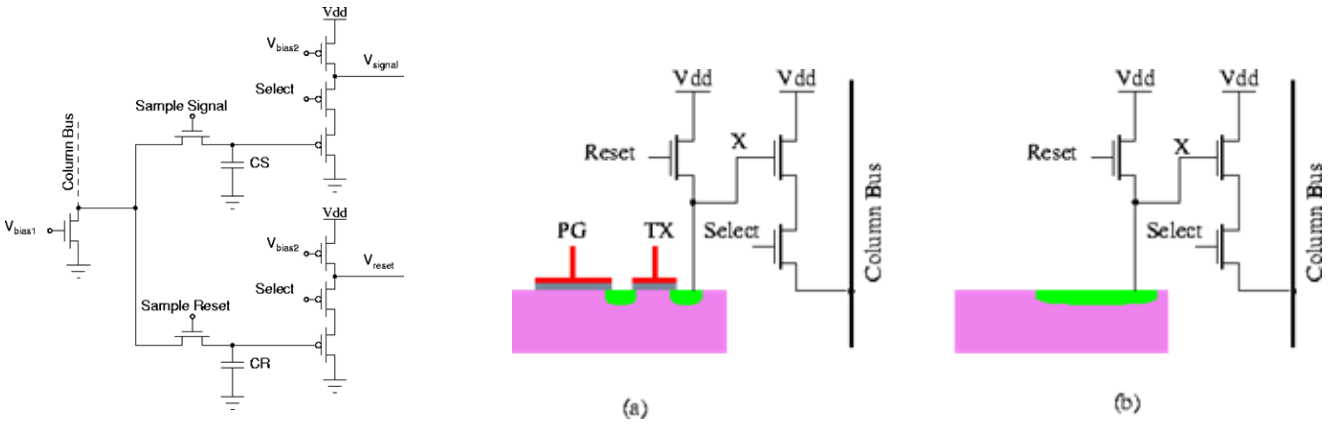


Figure 2.3.1-3 Schematic diagram of the correlated double sampling circuit (Left) and a) Active Pixel Array – APS photocircuit using a photogate. b) Using a photodiode (Right).

2.3.1 CCD image Sensors

Charge-coupled device – CCD is an architectural of image sensors where the charge collected by each pixel is transferred and converted to voltage levels, buffered, and sent out as an analog signal. This signal is then amplified and converted to numbers using an A/D-converter outside the sensor [2.5]. This technology was invented in 1969 in the Bell Laboratories by Willard Boyle and George E. Smith, who initially conceived the idea as a new type of memory circuit for computers. The image readout from a CCD consists of four stages where in stage 1 the exposure starts by the MOS Capacitors been positively charged and disconnected (bias). At stage 2 the shutter is open and electrons are generated from incident radiation for some time called integration time. When the integration time is over the stage 3 begins and the collected charges travel to the next capacitor with changes to the voltages of the capacitors gates. Electrons are attracted to nearest positively charged capacitor, which is biased and the degree of discharge is proportional to number of photons that hit each photodiode during the exposure. Finally at stage 4 the charge reaches the output convertor and then is amplified and converted into proportional voltage before send to A/D-converter to be digitalized [2.10].

The CCD architectures commonly used for high performance cameras are described below:

- Full frame CCD: Readout of the sensor takes pace when all rows are shifted one by one to a horizontal shift register and then amplified and converted by the last node. This is known as
  "Progressive Scan" readout. A full-frame CCD has the advantage of nearly all of its surface being photosensitive, with virtually no dead space between pixels. A disadvantage of full

frame is charge smearing caused by light falling on the sensor whilst accumulated charge signal is being transferred to the readout register. To avoid this, devices sometimes utilize a mechanical shutter to cover the sensor during the readout process. However, mechanical shutters have lifetime issues and are relatively slow. Shutters are not needed however in spectrographic operations or when a pulsed light source is used. Full frame CCD's are typically the most sensitive CCD's available and can work efficiently in many different illumination situations.

- Frame-transfer CCD: a two-part sensor in which one-half of the parallel array is used as a storage region and is protected from light by a light-tight mask. Incoming photons are allowed to fall on the uncovered portion of the array and the accumulated charge is then rapidly shifted (in the order of milliseconds) into the masked storage region for charge transfer to the serial output register. While the signal is being integrated on the light-sensitive portion of the sensor, the stored charge is read out. Frame transfer devices have typically faster frame rates than full frames devices and have the advantage of a high duty cycle i.e. the sensor is always collecting light. A disadvantage of this architecture is the charge smearing during the transfer from the light-sensitive to the masked regions of the CCD, although they are significantly better than full frame devices. The frame transfer CCD has the sensitivity of the full frame device but are typically more expensive due to the larger sensor size needed to accommodate the frame storage region.

- Interline-transfer CCD: Incorporates charge transfer channels called Interline Masks. These are immediately adjacent to each photodiode so that the accumulated charge can be rapidly shifted into the channels after image acquisition has been completed. The very rapid image acquisition virtually eliminates image smear. Altering the voltages at the photodiode so that the generated charges are injected into the substrate, rather than shifted to the transfer channels, can electronically shutter interline-transfer CCDs. Interline devices have the disadvantage that the interline mask effectively reduces the light sensitive area of the sensor. The transfer registers requires much space on the sensor which is not available for the actual image acquisition. The fill factor of the sensor is therefore only approximately 30 per cent. This can be partially compensated by the use of microlens arrays to increase the photodiode fill factor. The compensation usually works best for parallel light illumination but for some applications which need wide angle illumination (small F/# number) the sensitivity is significantly compromised [2.7].

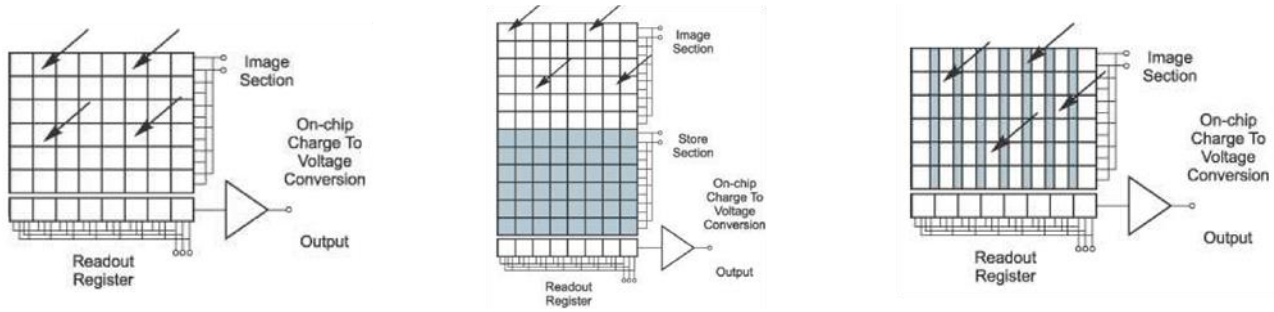


Figure 2.3.1-1 The three CCD architectures: a) Full frame, b) Frame-transfer and c) Interline-transfer

CCD imagers have the following performance characteristics in combination with the architectures above:

- Fill factor: The fill factor is basically the percentage of each pixel that is sensitive to light. Ideally, the fill factor should be 100%; however, in reality, it is often less than this as mentioned earlier. Features to control blooming (see below) and, for CMOS sensors, additional control electronics, occupy space within each pixel, and these areas are insensitive to light. The net effect of reducing the fill factor is to lower the sensitivity of the array.

- Dark current noise: Dark current can be defined as the unwanted charge that accumulates in the CCD pixels due to natural thermal processes that occur while the device operates at any temperature above absolute zero. At any temperature, electron-hole pairs are randomly generated and recombine within the silicon and at the silicon-silicon dioxide interface. Depending on where they are generated, some of these electrons will be collected in the CCD wells and will appear as unwanted signal charges (i.e. noise) at the output. The principal sources for dark current in order of importance are: generation at the silicon-silicon dioxide interface, electrons generated in the CCD depletion region, and electrons that diffuse to the CCD wells from the neutral bulk. The first two sources usually dominate the dark current. In addition, the generation rate can vary spatially over the array leading to a fixed noise pattern. For applications requiring very low noise levels, for example astrophotography, dark current sources can be reduced by cooling the CCD since they are strongly temperature dependent. The level of cooling is largely dependent on the longest integration time desired and the minimum acceptable signal-to-noise ratio.

- Quantum efficiency (QE): Quantum efficiency (QE) is the measure of the efficiency with which incident photons are detected. Some incident photons may not be absorbed due to reflection or may be absorbed where the electrons cannot be collected. The quantum efficiency is the ratio of the number of detected electrons divided by the product of the number of incident photons times the number of electrons each photon can be expected to generate. Visible wavelength photons generate one electron-hole pair, thus the QE for visible light is given by the ratio of the number of detected electrons divided by the number

of incident photons. There are a number of techniques used to improve the QE of CCDs, one of which is to illuminate the CCD from the back, as opposed to the front. In front-illuminated devices, incident photons must pass through the gate structure in order to generate signal electrons. Photons will be absorbed in these layers and thus won't contribute to the final signal. The absorption is also wavelength dependent, with short wavelength photons being absorbed more than long wavelength photons. This effect results in poor blue and UV spectral responses. In order to increase the short wavelength response, a technique of thinning the silicon substrate has been developed (typically from 300 μm down to 15 μm). In this case, the CCD is illuminated from the back and thus photons do not have to pass through the front gate structure. It has taken almost a decade to perfect the thinning process, the main problem being non-uniform thinning, i.e. the corners were thinner than the center which leads to non-uniform response, the 'potato chip factor'.

- Saturation and blooming : Blooming is an effect that occurs when, during the integration period, a potential well becomes full of electrons; this is usually caused by the presence of a bright object in the scene being imaged (assuming that the overall exposure is correctly set). When a potential well overflows, the electrons flow into surrounding potential wells, thus creating an area of saturated pixels. If blooming isn't controlled, the resultant image will suffer from large over- exposed regions. The charge capacity of an image sensor can be limited by either the individual photodiode characteristics (pixels) or the CCD itself, and is defined by the maximum amount of charge that the image sensor can collect and transfer while still maintaining all of its design performance specifications. This capacity limit is termed the saturation charge level, and when this limit is reached, the pixel or CCD is described as being saturated [2.10] (see figure2.3.1-2).

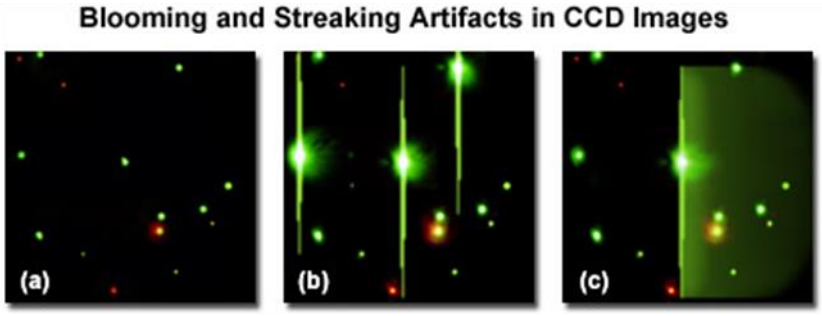


Figure 2.3.1-2 a) Normal image, b) image with blooming, c) image with blooming and saturation when maximum charge transfer capacity of the CCD is reached.

Many techniques have been developed to combat blooming, but one common method is to use lateral overflow drains – LODs (figure 2.3.1-3) .Conceptually, they work in a similar way to an overflow in a sink; when the potential well fills to a certain level, any further electrons that accumulate are allowed to drain away without affecting surrounding pixels. One of the drawbacks of using anti-blooming systems is the fill factor is often reduced, typically from near 100% down to around 70%. One result is low sensitivity in the near-infrared spectral region, due to the minimal charge generation near the CCD surface for longer wavelengths [2.9].

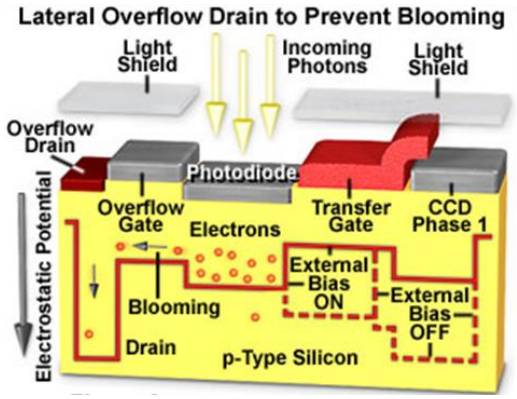


Figure 2.3.1-3 LOD allows electron overflow directly into the substrate rather than moving laterally into a drain channel

2.3.2 CMOS image Sensors

Complementary metal oxide semiconductor – CMOS, image sensors have been around for nearly as long as CCDs. These devices were made possible through research carried out at the Jet Propulsion Laboratory (JPL), and in 1993, they produced a CMOS sensor with a performance comparable to scientific-grade CCDs. Today CMOS imagers with CCD performance have made it to the consumer production line offering off the shelf products with outstanding performance. The continuous growth of CMOS sensors performance is the result of the industries trend to implement cheap and high performance cameras to anything from smartphones to automobile and aviation. The next step for CMOS imagers is to surpass CCD performance making them obsolete technology. An example from the industry is SONY which last August (2015) has discontinued the famous and very successful CCD products line[7]. In the next few pages we will review the function and performance of CMOS imagers to determine what makes them suitable for scientific grade imaging.

The main CMOS pixel architectures are presented and explained below [2.12, 2.13]:

- Passive Pixel Sensor – PPS: The PPS sensors are among the first CMOS imagers that emerged in the market. However, due to Signal to Noise Ratio (SNR) issues, its

[7] http://www.vision-systems.com/articles/2015/05/sony-finally-comments-on-ccd-image-sensor-discontinuation-news.html

development was halted. The working principle is the use of a single switch in the pixel to read out the photodiode integrated charge. The switch transistor in the pixel is used for the row selection (RS) and the output or the source terminal of this transistor is a common node for all the pixels in the same column. Because of this simple structure, this design reaches high fill-factor and photon collection efficiency. At the end of each column, there is a column select switch with horizontal access circuit. The design also includes an off-chip amplifier and Ana log to Digital Converter – ADC. However, PPSs suffer from many limitations. First, the common column line results in a high parasitic capacitance, which is also the charge to voltage conversion region of the pixel. This high parasitic capacitance is required to be driven by the pixel, which reduces pixel's conversion gain. Moreover, the off-chip read-out scheme introduces high noise and slow read-out. Later, PPS designs with on-chip amplifiers and ADCs have been introduced. These designs have reached lower noise in the sensor. However the high parasitic capacitance issue at the common column node still remains.

- Active Pixel Sensor – APS: In Active Pixel Array each pixel has an active element providing pixel-level amplification and improving the image quality compared to PPS designs. So an active pixel contains a photodiode, three transistors and four interconnects. The operation of this pixel is as follows: After a pixel is addressed by opening the row-select (RS) transistor, its charge is sensed and fed to the column bus by the source follower, afterwards the pixel is being reset by the reset (RST) transistor. The introduction of an amplifier in each pixel increases the non-uniformity between pixels analog nature resulting in the fixed- pattern noise effect. Also APS CMOS suffer from a large reset noise component.

- Digital Pixel Sensor – DPS: Digital Pixel Sensors are the CMOS sensors which have pixel level analog to digital conversion. With fabrication processes advancing the DPS design can be easily integrated and optimized. The DPS designs require large amount of non-photosensitive/metallic area causing a significant drop in the fill-factor and increase in the pixel pitch (how close can the pixels be). However, DPS architectures offer several advantages over analog image sensors including better scaling with CMOS technology due to the reduced analog circuit performance demands, and the elimination of read-related column Fixed Pattern Noise – FPN and column readout noise. More details on the DPS architectures will be covered in Chapter 5.2.1 where the sensor implemented to the design use column correction algorithms to compensate with FPN. The DPS designs may also serve well in microscopic applications when integrated with Back Side Illumination – BSI technology to compensate the low fill-factor and high metal density. The stacked CMOS technology can be helpful for higher integration levels. Both BSI and stacked technologies will be reviewed in more detail at the next topic consisting new technologies for CMOS imagers.

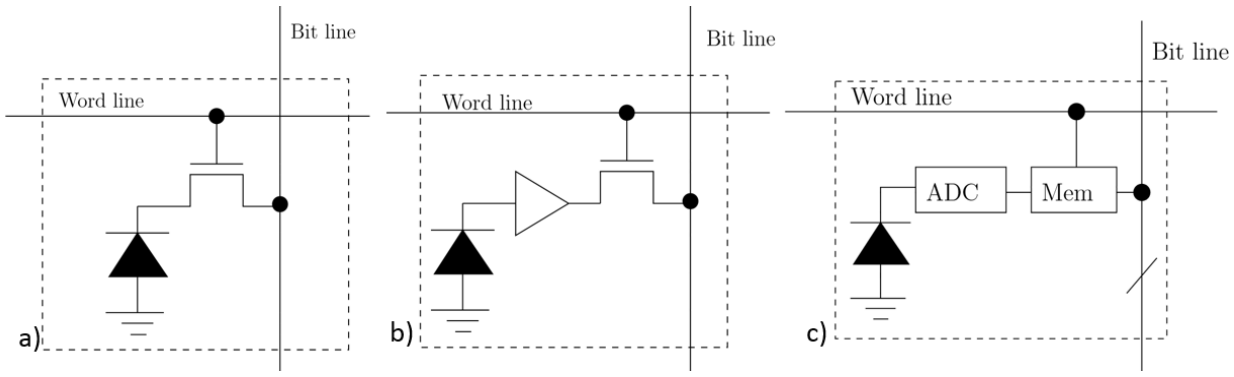


Figure 2.3.2-1 The three main CMOS architectures: a) Passive Pixel Sensor – PPS, b) Active Pixel Sensor – APS, c) Digital Pixel Sensor – DPS

Comparing the above architectures its worth mentioning that the most common design is the APS in combination with a pinned photodiode – PPD pixel structure. This pixel structure have made CMOS sensors competitive and reduced noise significantly [2.14]. In 1993, a CMOS active pixel image sensor (APS) with intra-pixel charge transfer was proposed by Fossum at JPL [2.15, 2.16]. Performance improvement using Backside Illumination – BSI and a pinned photodiode was suggested in 1994 [2.17]. A CMOS APS pixel with a PPD is shown schematically in figure 2.3.2-2. Signal charge collected by the pixel photodetector is transferred to a floating diffusion (FD) whose potential is monitored by a source-follower – SF within the pixel. FD is reset by transistor reset signal – RST prior to transfer and the source-follower is connected to the column bus line (COL BUS) using a row-select transistor (SEL). Since the PPD is often used in pixels with nominally four transistor gates, such a CMOS APS pixel is often referred to as a "4T" pixel. (This is in contrast to a "3T" pixel which refers to CMOS active pixel sensors where the photodiode is directly connected to the in-pixel source-follower, and complete intrapixel charge transfer from the photodiode is not performed. Sometimes "partially pinned photodiodes" [2.18] were used in 3T CMOS APS devices.)

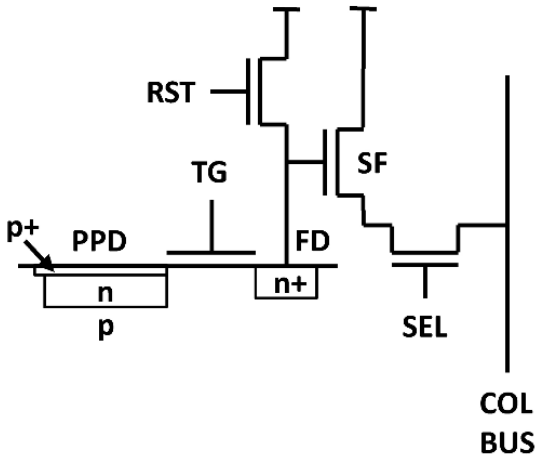


Figure 2.3.2-2 The PPD integrated to an APS structure

2.3.3 Imagers Integration technology

Front Side illuminated – FSI technology is the most mature and more common to entry level imagers. The structure of pixels is the simple PPD we have reviewed and the advancements in this technology include a hybrid CCD in CMOS sold under the name "sCMOS"(Scientific CMOS) consists of CMOS readout integrated circuits – ROICs that are bump bonded to a CCD imaging substrate. This technology that was developed for infrared staring arrays and now adapted to silicon-based detector technology (figure 2.3.3-1).

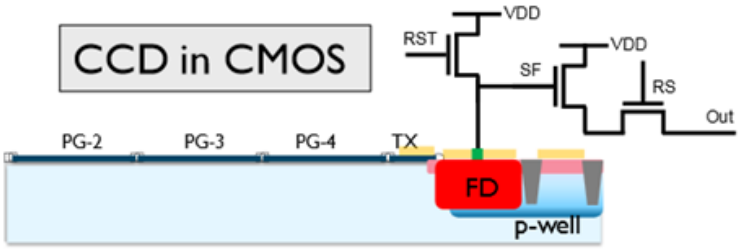


Figure 2.3.3-1 The sCMOS developed by Imec

Some of the applications of sCMOS are: Live cell microscopy, Solar astronomy, Bio- & Chemi – luminescence, Biochip reading, Hyperspectral imaging and more.

More information at http://www.scmos.com/

Back-Side illuminated sensors have the advantage in performance over FSI. For many years BSI imagers was used in scientific grate cameras and other high end applications like astronomy and military. For the past few years advancements consisting the easier manufacturing process have made the BSI sensors available to the consumer. The concept behind BSI technology is the thinning of the substrate leaving it only a few μm. The substrate is now thin enough for radiation to pass to the photodiode without the loss of photons. The result is far better QE than FSI. FSI imagers have great losses in QE because of the many layers above the photodiode that drive the pixel values out of the pixel array. BSI imagers have also better fill factor because of better angular coverage, lower noise because of limited optical and electrical crosstalk in comparison with FSI (figure 2.3.3-2).

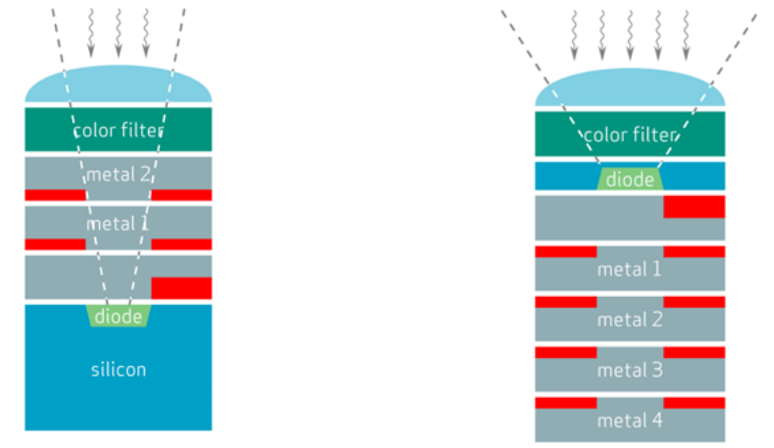


Figure 2.3.3-2 FSI pixel on the left and the new BSI pixel in the Right

3D stacked imagers are the next integration level for digital sensors. They enable a multi-layer design featuring not only a BSI design but also many layers of logic where many functions of the sensor like auto-exposure, auto-Whit balance or even a better auto-focus algorithms can take place. The logic enabled by such approach can be exploited in many ways to further improve image quality and add some smart features to the sensor. One interesting feature is the parallel readout of all pixels enabling a Global Shutter function. Global shutter can eliminate the artifacts of Electronic Rolling Shutter – ERS by reading all the pixels at once. Global shutter already exist to non-stacked sensors but is rarer. In more advance applications an interlayer cooling can be implemented enabling the more efficient cooling of the imager and significantly lowering thermal noise [2.19, 2. 20].

2.3.4 CCD and CMOS image Sensor Comparison

The final decision of the imager suitable for a new imaging platform for scientific purposes is the result of careful market research about the availability of many products and the performance they have to offer. In the previous topic the characteristics of both CMOS and CCD have been carefully explained. With the advantages and disadvantages of each technology in mind we are going to compare both for development and integration, power consumption, and cost / availability and also with image quality characteristics in mind like Quantum efficiency – QE, Spatial Resolution, Dynamic Range – DR and Noise.

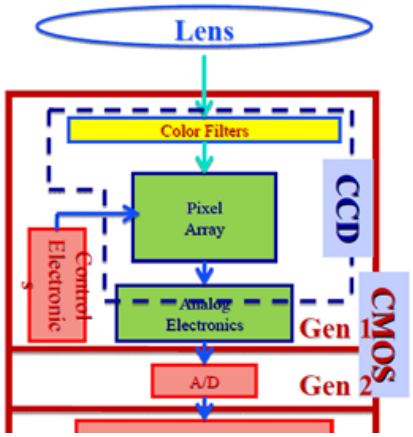


Figure 2.3.4-1 Integration potential of the CCD technology and CMOS technology

- Development and Integration: CCD require more circuitry outside the chip to function than CMOS. More precisely the voltages that move the charge in pixels need to be supplied by different phase clocks. The integration level is also smaller, featuring only the pixel array. CMOS imagers on the other hand have integration level that gives the final digital signal without the need of any electronics interference (figure 2.3.4 on the right).

- Power consumption: As stated above the CCD sensors have a lot of circuitry off the chip and for the correct charge transfer a multi-phase clocking needed. This approach is far more power consuming that the System on Chip approach of a CMOS sensor. A similar performing CCD can consume up to 100 times more power than the CMOS equivalent.

- Cost and availability: The final consumer cost has to be considered in combination with manufacturing cost. Large chip manufacturers develop inexpensive and off the shelf sensors only when in demand. The industry trend to produce high end image sensors for smartphones have dominated the need for new imaging technologies. In combination with CMOS capability for high integration level, eliminating the need of extend and more power consuming circuitry, as stated above, the CMOS sensor can be considered more value for money approach. On the other hand if a more specialized imager needed for some application, the cost of a custom CMOS design quickly rises and a better performing and less expensive CCD imager can be found. The availability of solid state imagers became an issue with manufacturers suppling only in large quantities and refusing to share technical details outside the business context. Fortunately some third party manufacturers can supply in small numbers or just one piece with technical details.

Image Quality

- Quantum efficiency – QE: Refers to the fraction of photons incident on the detector surface that actually generate electrons. The low QE of earlier CMOS devices is a direct result of pixel complexity. With several transistors in each pixel absorbing photons, the area of each pixel available for photodetection is well below 100%. The new CMOS sensors uses physically smaller transistors and related features, as well as a microlens array for light concentration, resulting in a fill factor and QE similar to CCDs.

- Spatial Resolution: The spatial resolution of an image can be limited either by the pixel spacing (pixel size or pitch) of the image sensor or the optical resolution of the imaging system (see Chapter 2.2). For the image sensor the pixel quantifies the photons reaching the photosensitive surface. In an ideal imaging system the sensor will have pixels that receive uniform photon distribution in each pixel because of the optical system resolution in a given radiation wavelength, but can distinguish between two of them the potential intensity difference. CMOSs and CCDs imagers both have new Backside Illuminated fabrication nodes capable of producing pixels small enough for almost any optical system. For Front Side Illuminated – FSI the fill factor is the only limitation (because of the non-photosensitive components covering the die) but within the limits of any decent optical system.

- Noise: The noise in CCDs and CMOS is very different [2.12]:
    - CMOS: The noise sources in CMOS image sensors are divided into three: temporal noise, input-signal noise, and spatial noise.

      The time-dependent pixel noise is called temporal noise and it includes the following:
      1- reset or kTC noise
      2- thermal noise or Johnson noise
      3- 1/f noise or flicker noise
      4- dark current shot noise
      5- photon shot noise
      5- quantization noise
      6- phase noise or timing jitter

      The temporal noise is "frozen" as spatial noise when a snapshot is taken. The spatial noise sources include:
      1- dark Fixed Pattern Noise – FPN
      2- light FPN
      3- leakers or hot spots
      4- defect pixels
      5- cosmetic defects (pixel-to-pixel non-uniformity) and these noises are fixed per pixel, per column or per chip.

    - CCD: As opposed to CMOS sensors, CCD sensors only include dark signal and shot noise. The read-out noise is not a part of the CCD sensor but is a part of the CCD camera.

  For a fair compaction the application of the imager can change the noise levels. For example for short exposures, read noise dominates the noise in the images and for longer exposures, thermal noise usually becomes the dominant factor. The dominant technology in this field can by decided only based on the application of the imagers.

- Dynamic Range – DR: Dynamic Range is the ratio between full well charge to the noise connected to this charge. In other words it describes the actual intensity levels that an imager can represent in proportion with incident photons integration (how much electrons the imagers pixel well can hold). This value depends on the pixel well capacity (well depth) and the total amount of Noise (readout and thermal) [2.1]. The noise play an important part in DR with pixel size giving a compensation. However, by staying within the limits of the diffraction, the largest possible pixel should be implemented for increased well capacity, which is important for both maximum SNR and larger DR. In some FSI CMOS the well capacity is limited due to the components in the pixel area

giving an advantage to CCDs. In BSI fabrication the pixel well of CCD and CMOS can have the same size and capacity leaving only noise to determine the superiority of each imager (as the technology itself offers no conclusive result).

Besides image quality and the other aspects of each imager technology examined there are some features and functional characteristics enabled by each technology:

CMOS can read any part of the pixel array without the need of the readout of the full pixel array like a CCD. In CCD the charge collected in each pixel is shifted and then collected without the ability to choose the area to be read. In CMOS sensors a Region of Interest can be pre-determined in order to be read as a frame featuring fewer pixels [2.21]. The readout of fewer pixels results to higher frame rates (see Chapter 7 for the Human Vision application).

The conclusion for the best suited technology for a wide range application smart and reconfigurable imaging system capable of interfacing with various imagers concludes to CMOS image sensors. One aspect that is not mentioned in the performance comparison is the many flavors of CMOS imagers available. Imagers with Near Infra-Red – NIR sensitivity and more specialized characteristics are available only for CMOS technology.

2.4 Frame Acquisition

Frames are still images in a video captured by a digital image sensor. More precisely are the information read from the pixels each time and presented until the next readout takes its place. Imagers give out some control signals with the pixel values to indicate the frame structure and limits. In a parallel data readout interface there are the parallel data indicating the pixel value (for an 8bits sensor there are D0 to D7 out pins) and some control signals. The more common control signals are the Pixel clock that indicates the period that the Data values are valid for the specified pixel, the Line Valid or Horizontal Sync for letting the Image Signal Processor – ISP know that the line of pixels that it received was valid and for last the Frame Valid or Vertical sync indicates the start and end of a frame. A visualization of this scheme is shown in figures 2.4-1 and 2.4-2.

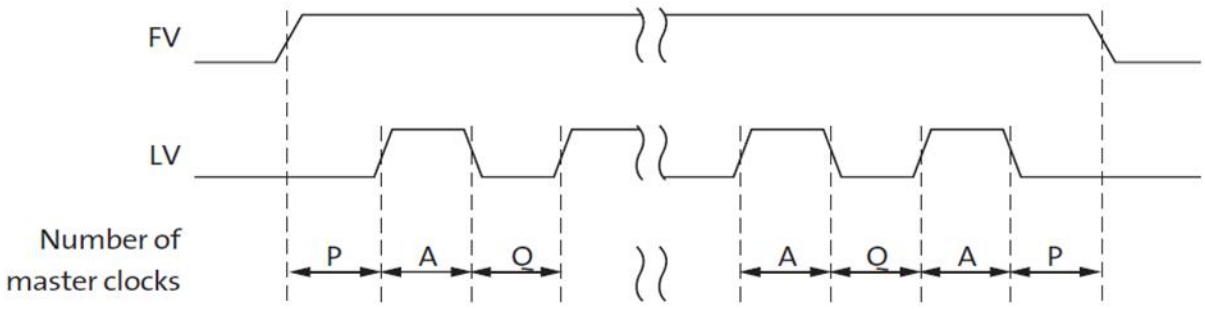


Figure 2.3.4-1 The Frame Valid – FV and Line Valid – LV signals indicating the valid frame pixel readouts. The frame stars after the first logic true in FV&LV (first A). After the desertion of LV the first line of the frame has arrived. The process is repeated until

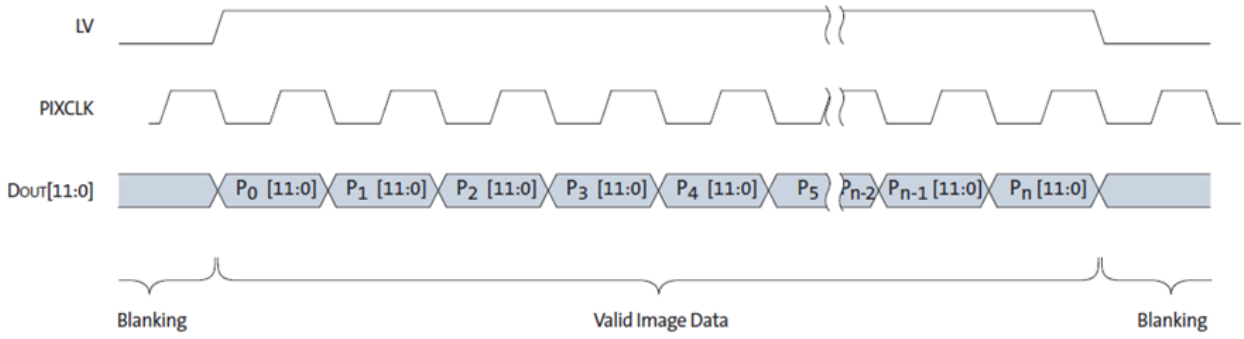


Figure 2.3.4-2 In more detail this figure shows the data readout from a sensor with 12 bit parallel interface. The valid pixel values are indicated by the Pixel Clock – PIXCLK

The application processor or ISP uses these control signals to create the frame in a frame buffer or to transmit it with a protocol in a way to indicate the start and end of the frame (see Chapter 2.6).

The extended number of output wires required for a parallel interface like the one described above and the limitation they suffer has led the manufacturers to serial interfaces. More precisely the extended number of pads in an Integrated Circuit – IC case parasitic capacitance lowering the speed the circuit can operate. Alternately a chip with parallel interface consumes more energy and transmits fewer data. With sensors featuring higher resolution the parallel data interface is no longer able to handle the bandwidth requirements [2.22]. The Low Voltage Differential Signaling – LVDS protocol was introduced as a solution to the limited bandwidth presented by parallel interfaces. National Semiconductor first introduced Low Voltage Differential Signaling as a standard interface device back in 1994. National realized that the demand for bandwidth was increasing at an exponential rate while users also desired reductions on power dissipation [2.23]. Different manufacturers refer to their versions of LVDS with different names (subLVDS for SONY, HiSPI for ON Semiconductor etc.). The basic principle behind LVDS is that the data are serialized and transmitted thru two opposite biased wires (figure 2.4-3). The data are recovered at the end where the lines are terminated by operational amplifiers. The difference in the voltage of

the two wires is the recovered data. This scheme offers noise immunity because of the full associativity in the noise presented in the two wires. Electro-Magnetic Interference – EMI is also rejected from the subtraction of the two signals making them ideal for fast and low voltage applications. The extend bandwidth that LVDS offers comes with minimum power consumption (Gigabits @ milliwatts) making it perfect for low power consumption applications like mobile devices. The Mobile industry Processor interface – MIPI Alliance have release a very well documented protocol based on LVDS called Camera Serial Interface – CSI2 [24]. CSI2 is very common in many modern imagers.

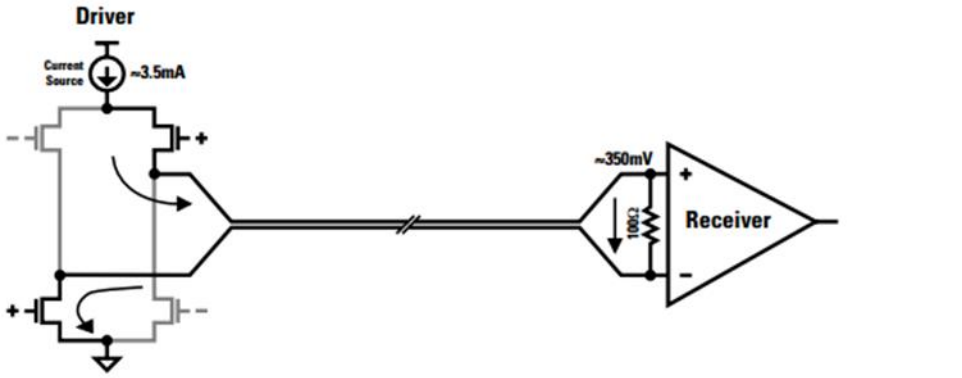


Figure 2.3.4-3 The LVDS Data lane with a driver and a terminal receiver.

After examine the available imagers with parallel and serial support it was clear that the support of a serial interface was mandatory. In an exhaustive market research the result for the most advanced and futureproof imagers was from Sony, supporting subLVDS. One manufacturer offering the same protocol was Panasonic with limited number of imagers. In the scope of the theoretical background explanation we will discuss only the deserialization of the date from serial to the parallel interface explained above.

Deserialization of data can be done with several hardware choices like Application-Specific Integrated Circuit – ASIC or Field Programmable Gate Arrays – FPGAs. The best choice after research was the MACHXO2 FPGA from Lattice. The MACHXO2 FPGA can convert HiSPi-to-parallel and subLVDS-to-parallel with different IP cores (figure 2.4-3).

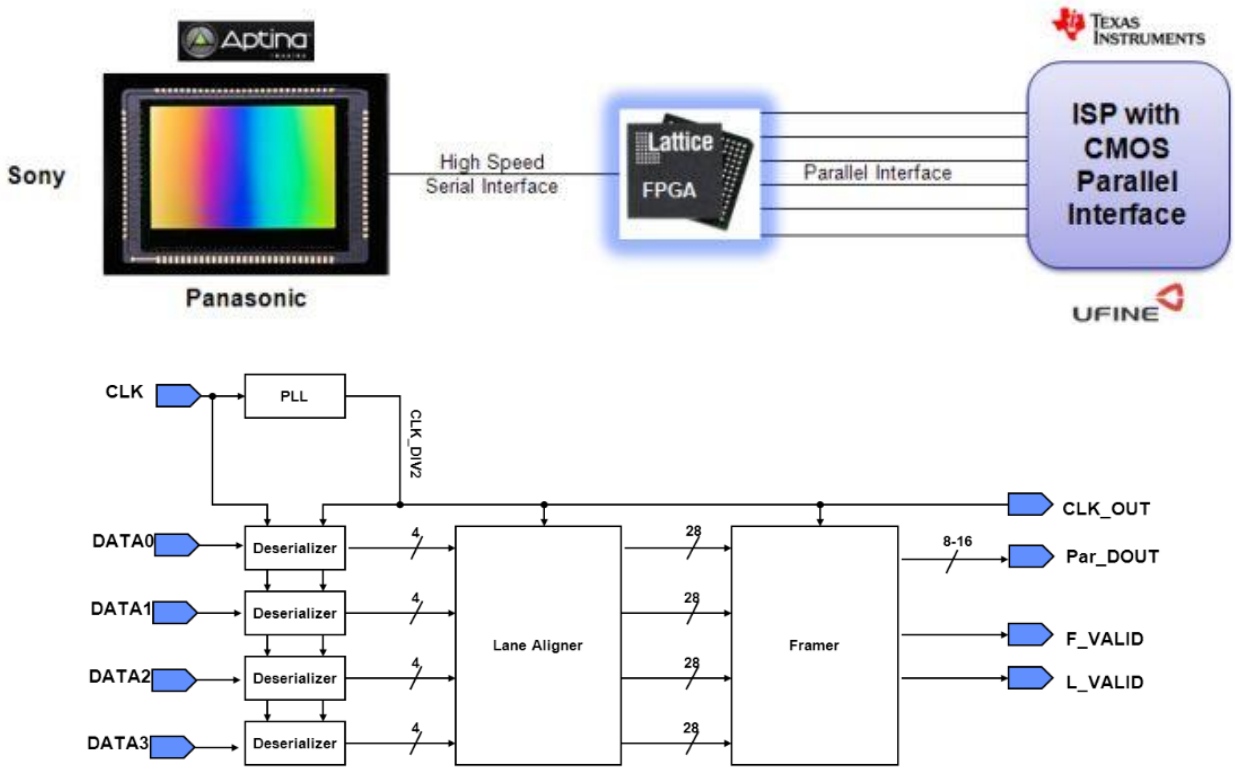


Figure 2.3.4-4 The Bridge converting a serial interface to parallel above and the IP core enabling deserialization of 4 lanes of serial data below.

2.5 Image Processing

After the frame is constructed and saved into a memory, the image processing algorithms need to reconstruct the color from the pixel intensities and make a tone mapping to address the dynamic range of the reconstruction terminal. There are two possible ways common to digital imagers to recognize color. A beam splitter with a dedicated imager for each band and a Color Filter Array – CFA. The two methods are demonstrated in figure 2.5-1 below.

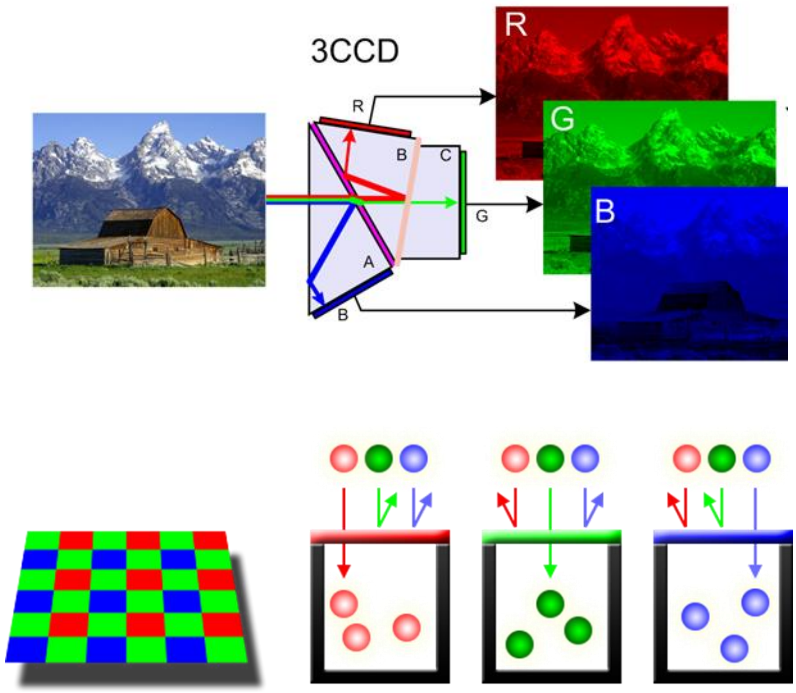


Figure 2.3.4-1 Above the beam splitter color separation with A, B and C CCD detectors and below a Bayern CFA .

The CCD sensors are commonly used with a beam splitter to separate each color band. The working principle behind color separation is the different reflectance index of each wavelength. One big disadvantage is that they need to be aligned perfectly in order to give a correct image, which makes this technology very fragile. If not aligned correctly the result is similar with the chromatic aberration in optics (Chapter 2.2). The CFA on the other hand is more commonly used in imagers. Each color is passing thru a bandpass filter which lets only the photons of the wavelength around Red, Green and Blue for each filter to pass thru. The actual pixel presented in the final image has 3 values (Red, Green, and Blue) which are reconstructed with a demosaicing algorithm.

Demosaicing is the procedure where the missing color values are calculated by the values of the neighbor pixels. This procedure is based in the fact that photons of a specific band are more likely very similar in the near proximity of the pixel been examined. This approach tends to cause blurring to the final image because of the inaccurate estimation in the missing pixel values. Different demosaicing algorithms have different performance in image quality but tend to be more complex and computationally demanding for video.

There are many CFAs resulting to different algorithms. The most popular CFA pattern is the Bayer pattern, which features blue and red filters at alternating pixel locations in the horizontal and vertical directions, and green filters organized in the quincunx pattern at the remaining locations [2.25]. Bayer CFA [28] is the optimal spatial arrangement in terms of spatial frequency representation when placing three colors on a square grid. Other proposed CFAs, such as stripes

or interlaced stripes, have at least the horizontal or vertical direction sub-sampled by a factor of three. Also, the four-color CFA proposed recently does not provide a reduction of aliasing since its frequency pattern is identical to the Bayer and the addition of a color should reduce correlation between color channels and worsen aliasing by increasing the chrominance bandwidth [2.27].

The error in estimated color values for each pixel introduce some artifacts resulting in image quality degradation. . In literature many demosaicing techniques have been proposed but the reconstructed images are affected by some visible and annoying artifacts [2.26]. The result of these artifacts is shown in figure 2.5-2.

Figure 2.3.4-2 Some Demosaicing artifacts.

Excessive blurring: Blurring is introduced when non-uniform color areas are mixed together by the estimation algorithm (upper Left picture).

Grid effect: Is introduced when a wide filter is used to extract the luminance component. The result is that chrominance components are interpreted as luminance components (Upper right picture). (In NTSC color television, a very similar phenomenon is called "chrominance-luminance crosstalk").

Watercolor: The "watercolor effect" is essentially a "smearing" of the chrominance. It results from using too narrow filter to extract the chrominance components. (Down – Left picture).

False color: This artifact typically manifests itself along edges, where abrupt or unnatural shifts in color occur as a result of misinterpolating across, rather than along, an edge. (Down – Right picture).

Other demosaicing artifacts presented in poor resolution images are shown in next page by figure 2.5-3.

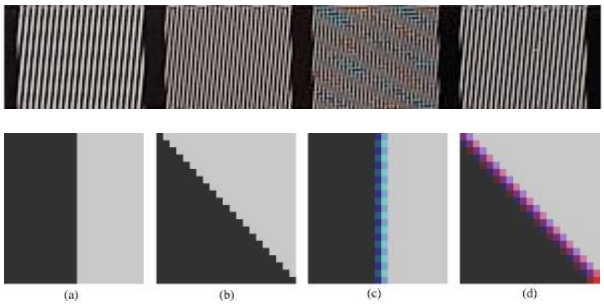


Figure 2.3.4-3 : The Aliasing artifact (above) presented when there is aliasing between the green, red and blue spectrums. The zipper effect is presented when pixels of a uniform area in image get mixed together by the demosaicing algorithm with another uniform area

The most common demosaicing algorithm is the Linear Interpolation – LI algorithm where a 3x3 neighborhood is taken from the CFA, and missing pixel values are estimated by the averaging of nearby values (interpolated). The algorithm is presented in four stages [2.29] shown below in figure 2.5-4:

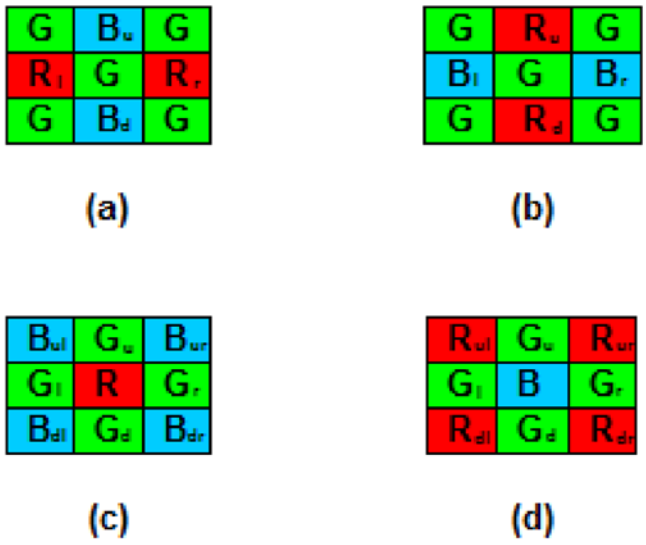


Figure 2.3.4-4 The four cases of pixel interpolation in NN algorithm.

To better understand figure 2.5-4 and the LI algorithm the four cases are better explained below:

a) In this case the pixel examined is a Green in the same line with Red pixels (see Figure 2.5-1). In this case the pixel examined will referred as Gr. The pixel above is a Blue and is referred as Bu. The same happens with the pixel below (Bd). Same principle is applied to the Red pixels left and right (Rl and Rr). The missing values are calculated as shown in the following equations:

$$R_{Gr} = \frac{R_l + R_r}{2} \qquad B_{Gr} = \frac{B_u + B_d}{2}$$

b) In the second case there is a Green pixel in the same line with blue pixels - Gb. The missing values are calculates as follows:

$$R_{Gb} = \frac{R_u + R_d}{2} \qquad R_{Gb} = \frac{B_l + B_r}{2}$$

c) This case involves a Red pixel – R which is surrounded always by the same type of pixels. The missing Blue values are calculated by the upper left, upper right, down left and down right Blue pixels.

$$G_R = \frac{G_u + G_d + G_l + G_r}{4} \qquad B_R = \frac{B_{ul} + B_{ur} + B_{dl} + B_{dr}}{4}$$

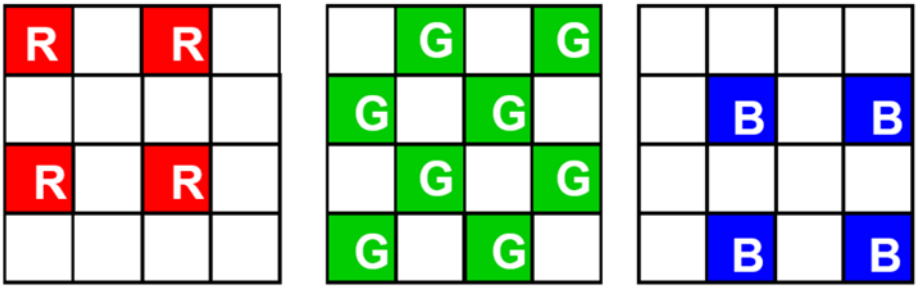
d) The final case involves a Blue pixel and the same principle as the Red pixel applies.

$$G_B = \frac{G_u + G_d + G_l + G_r}{4} \qquad R_B = \frac{R_{ul} + R_{ur} + R_{dl} + R_{dr}}{4}$$

This interpolation method performs well in smooth areas where the color changes slowly from one to the next. However, when performed along edges where color changes occur abruptly, false color and zipper artifacts are introduced, resulting in a poor image quality.

A more advanced demosaicing algorithm is the Bilinear Interpolation – BI algorithm. Bilinear interpolation is an extension of LI involving pixels from a bigger area. An example of BI algorithm is shown in figure 2.5-5.

$$R_F(n_1,n_2) = \mathbf{F}_R \otimes M_R(n_1,n_2)$$
$$G_F(n_1,n_2) = \mathbf{F}_G \otimes M_G(n_1,n_2)$$
$$B_F(n_1,n_2) = \mathbf{F}_B \otimes M_B(n_1,n_2)$$
$$\mathbf{F}_R = \mathbf{F}_B = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \Big/ 4 \quad , \quad \mathbf{F}_G = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \Big/ 4$$



$M_R(n_1,n_2)$ $M_G(n_1,n_2)$ $M_B(n_1,n_2)$

Figure 2.3.4-5 Bilinear interpolation with a 4 X 4 window

In this method a window containing the surrounding pixel values is convoluted with a weight kernel. This calculation can be done using a Fourier transform function (fft in matlab), resulting in a much faster execution for the incoming frames of a video. The results of BI algorithm are much better than LI consisting image quality but there are algorithms with better image results but more computationally demanding [2.30].

An offline algorithm called Contour Stencils – CS can be used in RAW saved data of an image to give the best image quality regardless performance [2.31]. The CS algorithm uses the Contour stencils method to calculate the contours of a uniformly sampled grayscale or color image. Mosaicked contour stencils are a medication of contour stencils for the Bayer pattern. The algorithm then proceeds to select the stencil according to the CFA of the imager. The CS algorithm has improved angular resolution because of a demosaicing method based on minimization. The demosaicing is performed by solving the constrained minimization shown below:

$$\begin{cases} \arg\min_{u} \sum_{m} \left(\sum_{n} \left(w_{m,n} \| u_m - u_n \|_L \right)^2 \right)^{\frac{1}{2}} + a \sum_{m} \left(\sum_{n} \left(w_{m,n} \| u_m - u_n \|_C \right)^2 \right)^{\frac{1}{2}} \\ subject\ to\ u_m^k = f_m, m \in \Omega^k,\ k \in \{R, G, B\} \end{cases}$$

Where:

- f denotes the observed mosaicked image,
- u is the demosaiced image that is found by the minimization,
- Ω^k denotes the set of pixel locations where color component k is known, k ∈ $\{R, G, B\}$
- a is a positive parameter,
- $L\ and\ C$ indicate seminorms in the color space

Demosaicing involves a balance of avoiding two deferent undesirable effects, color artifacts and zipper artifacts (described above). In contour stencil demosaicing, the parameter a explicitly controls the balance between color artifacts and zipper artifacts. This parameter is very useful in different kind of images as discovered in some applications of the system.

2.6 Video Representation

Transferring and reproducing video has many aspects consisting compression color space and different protocols. In the scope of this thesis the YUV and RGB color spaces will be discussed.

2.6.1 YUV Color space

The YUV color space involves 2 components, Chrominance – Croma (U and V) and Luminance – Luma (Y). This representation of images is very common in image pipelines and ISPs. In earlier

years the YUV was used by the PAL (Phase Alternation Line), NTSC (National Television System Committee), and SECAM (Sequentiel Couleur Avec Mémoire or Sequential Color with Memory) composite color video standards [2.33]. This way it was possible to transfer black and white video using only Luma (Y) information; color information (U and V) was added in such a way that a black-and-white receiver would still display a normal black-and-white picture[8]. This principle is possible to be used by a modern imaging system to transfer pixel intensities without the need of conversion or specialized machine drivers (see chapter 5). One more advantage of YUV other than historical reasons is that the human eye is less sensitive to changes in hue than changes in brightness [2.37]. As a result, an image can have less chroma information than luma information without sacrificing the perceived quality of the image. For example, it is common to sample the chroma values at half the horizontal resolution of the luma samples. In other words, for every two luma samples in a row of pixels, there is one U sample and one V sample (subsampling of the chroma component) [2.32].

One version of subsampled YUV is the YUV2. In the YUV2 the Luminance values Y are sampled for every pixel, but chrominance values U and V are sampled only for even pixels:

Y0, U0, Y1, V0 (first 2 pixels)

Y2, U2, Y3, V2 (next 2 pixels)

Y4, U4, Y5, V4 (next 2 pixels) . . .

The result is 25% less information to transfer with a good final result. If the luminance component is used to transfer the pixel intensities and the chrominance component U and V is in a continuous fixed pattern (hard-coded) then the final result will be the image data without any loss of information and image quality.

2.6.2 RGB Color space

The simplest and most commonly used protocol is the RGB. It consist of three bands corresponding to the wavelength of Red, Green and Blue. The combination of the three bands can make any color that the human eye will see. The choice of basic colors is related to the physiology of the human eye; good primaries are stimuli that maximize the difference between the responses of the cone cells of the human retina to light of different wavelengths, and that thereby make a large color triangle (see figure 2.6.2 below).

[8] https://msdn.microsoft.com/en-us/library/windows/desktop/bb530104%28v=vs.85%29.aspx

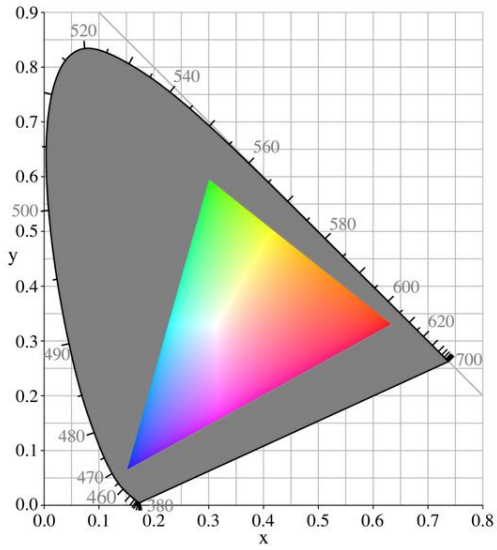


Figure 2.6.2-1 The sRGB color triangle, shown as a subset of x,y space, a chromaticity space based on CIE 1931 colorimetry

Transition of frames in the RGB protocol require a fully demosaiced image with all the information in the red, green and blue bands. The band components have the necessary information to present a color image in full resolution without subsampling. For example in RGB24 there are 3 Bytes of information for each pixel. If only one band is used to transfer the pixel intensities it will result in only 33% usage of the available bandwidth. To take advantage of each band each band must carry out some information about the pixel intensities. One way to do that is (for a 24 bit sensor) to send the 8 Least Significant Bits – LSB in the R component, the next 8 bits in G component and the rest Most Significant Bits – MSB to the B component. The recovered band information is then concatenated and the result is the initial 24 bits of the RAW pixel values captured by the imager.

2.7 Imaging System Performance

As mentioned before in Chapter 2.2, the image resolution of an optical system is determined by the angular resolution of two distinct objects. For an imaging system a good imaging performance metric is the Modulation Transfer function – MTF [2.35]. In the next few pages the theory behind MTF and imaging performance tests will be explained in detail.

2.7.1 Modulation Transfer Function – MTF

MTF presents the spatial frequency response of an imaging system or a component of the system and it is the contrast at a given spatial frequency relative to low frequencies [2.36]. To fully comprehend the meaning of this sentence, the concept of resolution and contrast will be explained. The aspect of an object's image transferred from object to a plane is also mandatory and although it has already been explained in great detail by Chapter 2.2, the MTF aspect of the optical system is not covered and will be examined by the next paragraphs.

MTF in a more simple way represents the measure of the transfer of modulation (or contrast/sharpness) from the object to the image. In other words, it measures how accurate the system reproduces (or transfers) detail from the object to the final image. It is generally expressed as the ratio of the relative image contrast divided by the relative object contrast:

$$MTF = \frac{Relative\ Image\ Contrast}{Relative\ Object\ Contrast}$$

The sharpness of a component of the system, or the full system sharpness can be characterized by the MTF, and is also known as Spatial Frequency Response – SFR.

Spatial frequency is typically measured in cycles or line pairs per millimeter (lp/mm), which is analogous to cycles per second (Hertz) in audio systems [2.36]. In comparison with the frequency of sound, which is perceived as pitch and measured in cycles per second (Hertz) audio components (amplifiers, loudspeakers, etc.) are characterized by frequency response curves. MTF is also a frequency response, except that it involves spatial frequency - cycles (line pairs) per distance instead of time.

Spatial Frequency Response – SFR is the contrast at a given spatial frequency relative to low frequencies. High spatial frequencies correspond to fine image detail. The response of the system components (imaging sensor, lenses, filters, etc.) tends to roll off at high spatial frequencies. These components can be thought of as lowpass filters (filters that pass low frequencies and attenuate high frequencies).

Another approach is that MTF corresponds to the bandwidth of a communications system; homogenous pixel areas that should be different (sharper) by the objects image corresponds to its noise. This pixel areas can be characterized by a frequency spectrum (higher frequencies correspond to finer pixel patterns) as well as amplitude (intensity or contrast). It can be hypothesized that the Shannon information capacity of an imaging system (a function of bandwidth and noise) correlates with perceived image quality.

The components that characterize MTF, as mentioned earlier, can be described as resolution and contrast. Resolution and contrast are inseparably bound. Think of a series of alternating lines; black, then white, then blank, then white again. What differentiates them is their contrast. If we

make the black lines lighter and the white ones darker then eventually they merge. In other words, if a white line is present on white paper then there is no contrast and therefore no resolution. Therefore a discussion of resolution without taking contrast into account is meaningless.

- Resolution: The concept of resolution of an imaging system is the system's ability to distinguish object detail. Resolution is usually measured in lines per millimeter, or line pairs per millimeter. (L/mm and LP/mm). Typically scientific types refer to lines per millimeter and lines per millimeter are used by photographers. The measure of line-pairs per millimeter (lp/mm) is also known as frequency. The inverse of the frequency yields the spacing in millimeters between two resolved lines. Below in figure 2.7.1-1 there is an example where we have maximum contrast but resolution failure when the resolution is quantified by the sensor even if the line pair of the image in the photosensitive surface is the same line pair.

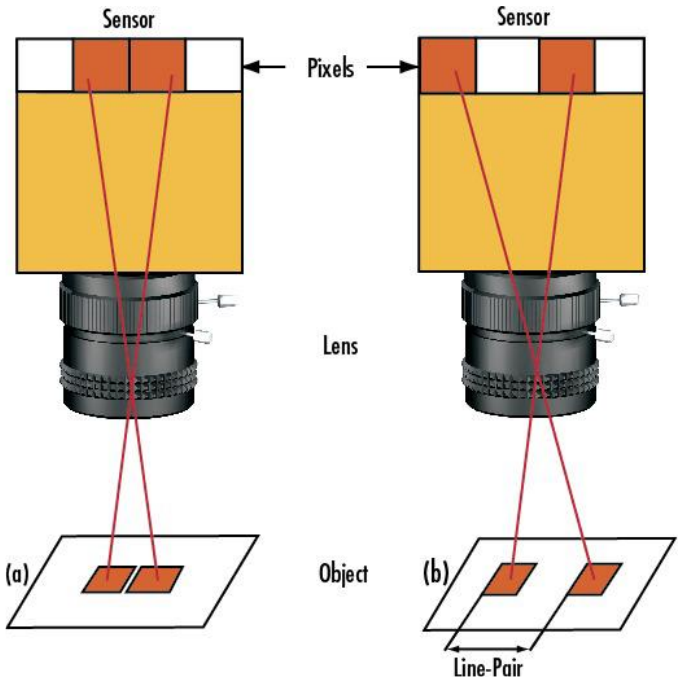


Figure 2.7.1-1 Imaging Resolution Scenarios where (a) the line-pair is not resolved and (b) the line-pair is resolved [35 – page 74].

The MTF resulting by a system with generally good contrast but poor resolution is presented in figure 2.7.1-2 [2.35 – page 73].

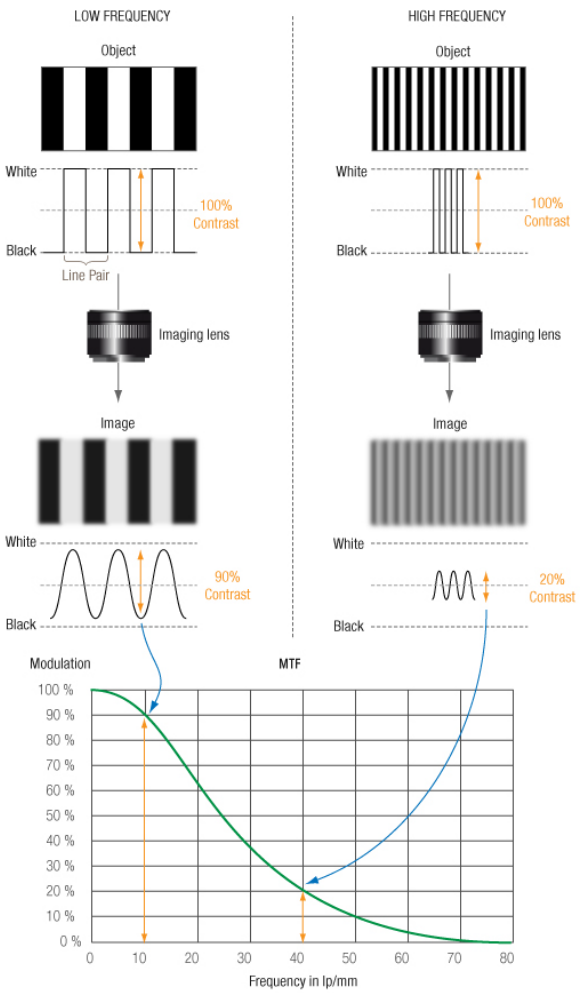


Figure 2.7.1-2 The MTF of a system at a low and high frequency and the MTF curve

The MTF curve above shows how spatial resolution affects contrast. The x-axis represents the spatial frequency in line pairs per mm and the y-axis (MTF value) represents the contrast restitution for the corresponding spatial frequency. The maximum attainable frequency on the sensor is called the Nyquist frequency and corresponds to alternating dark and bright lines one pixel wide. The MTF value shows the restitution of the frequency at each percentage. The value of the MTF at frequency 0 is always 100% since a flat field is considered to have been reproduced perfectly, with no intensity loss [2.35].

- Contrast /Modulation: In a perfect line pair the white square has the maximum possible intensity value and the black one absolute zero. Assuming perfect resolution in ideal conditions there is no blurring between them and the line pair is perfectly distinguishable. The next line pair is assigned to intensity values closer together resulting to a "darker" white with lower intensity values and more "whitened" black assigned to a higher value equal to the difference between the initial maximum value of the perfect white and the smaller value of the "darker" white of the next line pair (figure 2.7.1-3). The mere line pairs we create by following this approach the worst contrast we have. At the end there will be a line pair which the black and white lines will have the same intensities. This is an example of contrast failure in the best possible resolution. A sampler way is thinking of a white line drawn at a white background.

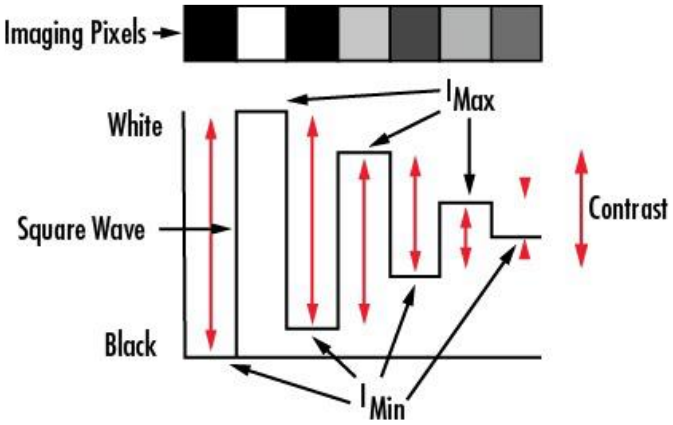


Figure 2.7.1-3 Contrast expressed as a square wave at different levels of resolution.

Mathematically, contrast is calculated with Michelson contrast equation:

$$Contrast/Modulation = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

Based on the above equation the MTF is given as:

$$MTF(f) = \frac{M_{captured}(f)}{M_{original}(f)}$$

The $M_{captured}(f)$ is the modulation of the captured image in a given frequency and $M_{original}(f)$ is the modulation of the actual object.

2.7.2 Resolution Test

In order to test this system performance the following test will be conducted

Grill/Square-Wave Analysis Pattern:

The target in this test to determine MTF involves the pattern shown in figure 2.7.1-4.

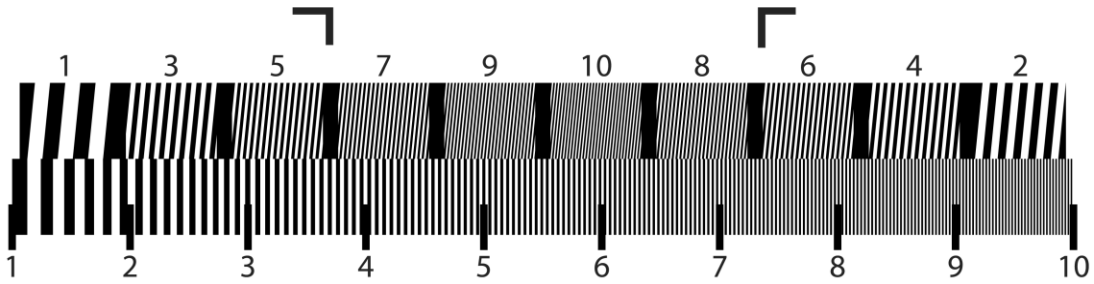


Figure 2.7.2-1 Grill Pattern Target

The square-wave analysis is performed by test patterns of consecutive bar patterns with augmented frequency. Such targets feature series of lines of specific frequencies that measure resolution by determining how far an imaging system can distinguish individual lines. The more different groups of variable frequencies in the test-target, the wider range of tested resolution the system reaches. This kind of target in a resolution chart is easy to produce and in combination with a software developed by Chantzi Efthymia in the scope of her thesis [2.35] the RAW MTF indicating the system performance can be done easily and accurately. More details about the test follow at Chapter 6 were the imaging system performance evaluation is conducted.

This software perform uses the direct square-wave MTF analysis [2.35, page 85].

3 RELATED WORK

The related work, similar to this thesis, will be divided into two sections. The first section includes the previous thesis done at the Optoelectronics and Imaging Diagnostics Laboratory by Katerina Trilyraki last year (July 2014) with the same objective. The second section I will preview a design very similar to the one done in this thesis using the older model of the controller I use.

3.1 Objective Oriented

The Development of an advance platform for controlling the imaging parameters of CMOS image sensors was done by Katerina Trilyraki [3.1] and was the first system developed able to control the imaging parameters of a CMOS image sensor. The platform used was the OMAP-L138 Low Cost Development Kit (OMAP-L138 LCDK) [3.2] (shown in figure 3.1-1). This processing module belongs to the category of heterogeneous multicore processors. It use a Digital Signal Processor – DSP and an ARM subsystem and associated memories (ARM926EJ-S) to process the frames captured by a peripheral called VPIF. The controlling of the CMOS sensor was successful with new smart features supported. Some smart applications like the Human Vision emulation, De-noising during Acquisition, Ultra-Fast Spectrometer and Pseudocolor Mapper showed impressive results and inspired me to continue my work and develop the current imaging system.

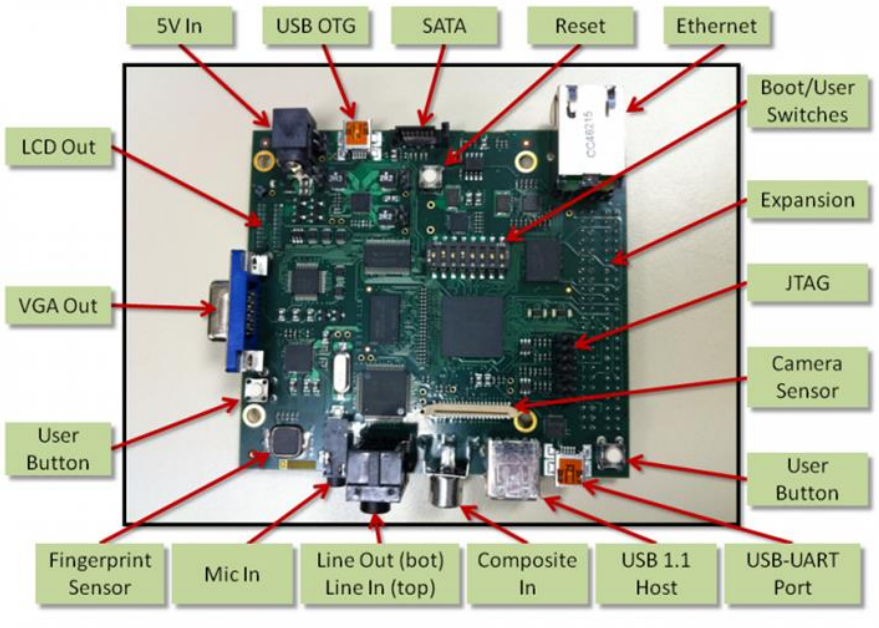


Figure 2.7.2-1 The OMAP-L138 Low Cost Development Kit (OMAP-L138 LCDK)

Nevertheless despite the ingenious implementation, with assembly optimizations to the algorithms running on both ARM and DSP, the performance results was disappointing. The frame rate was 26.3 fps for 640 * 480 (VGA) resolution and with the Nearest Neighbor demosaicing (a more simple and lower quality than linear demosaicing) algorithm resulting to the expected image quality. One more disadvantage was the system connectivity. The output used was an analog VGA with connection to only monitors. The dynamic range of the image captured was quantified with tone mapping to address the RGB565 color format for the VGA resulting in even poorer image quality. Although the system has various hardware to connect with a host like Gigabit Ethernet and a USB 3 port, the platform was already under computational stress, and with no operating system running the development of drivers for the peripherals was extremely difficult and unclear.

The image sensor module (LI-CAM-M034) used is the same I use for the current platform. The camera drivers needed some modifications to run smoothly to the new hardware, but it was very helpful having an example to follow.

3.2 Hardware Oriented

The example consisting similar hardware was done by Thomas Lundmark at the Lulea University of Technology, Sweden as a master thesis. The design involves the development of a USB 2 camera using a general purpose microcontroller [3.3]. A Printed Circuit Board – PCB was developed featuring the Cypress FX2LP microcontroller with USB2 connectivity.

The controller use an older version of the peripheral currently featured in this design called GPIF.

The image sensor used was the Cypress IBIS-B-1300 1.3 MP CMOS Image Sensor with full frame readout time at 36 ms (max. 27.5 fps).

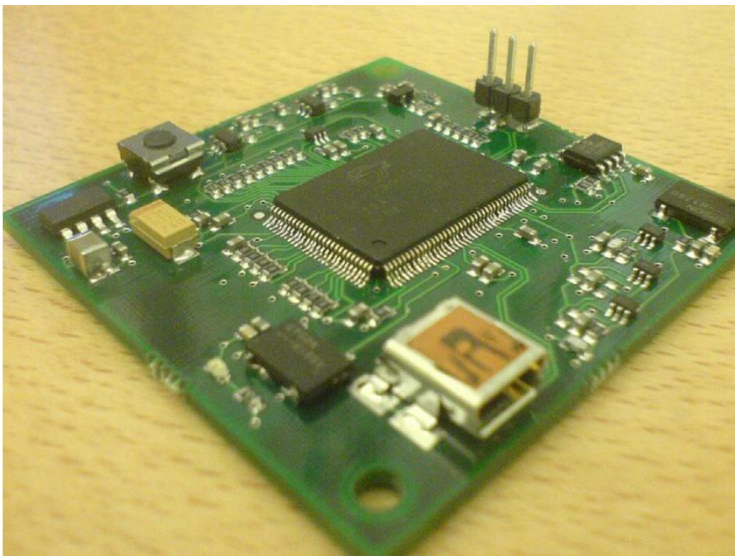


Figure 2.7.2-1 The PCB created with the FX2LP microcontroller.

4 PROPOSED SOLUTIONS

In this chapter we will review some of the proposed solutions suggested to address the problem of hardware capable of interfacing with different imagers and transferring frames with enough resolution and frame rate.

4.1 Multicore Embedded system-on-Chip – SoC

The continuous growth of computational power in ARM based systems makes them ideal for the task. Most modern mobile ARM based microprocessors future a powerful GPU like the NVIDIA TEGRA X1[9], a state of the art Soc capable of delivering 1 teraflop of computational power[10].

One disadvantage of these systems is the development support for new devices. Online help is limited and there are limitations to compatible drivers to expand connectivity.

A promising alternative was Exynos 5-based Arndale community board (figure 4.1), ARM Cortex-A15 dual-core CPU and ARM Mali T604 GPU[11]. It features connectivity for a MIPI CSI-2 (see Chapter 2.4) imager. As determined by the market research done consisting the CMOS sensors, the CSI-2 is not one that is supported by the high end imagers, making the Arndale board unsuitable for the goal set.

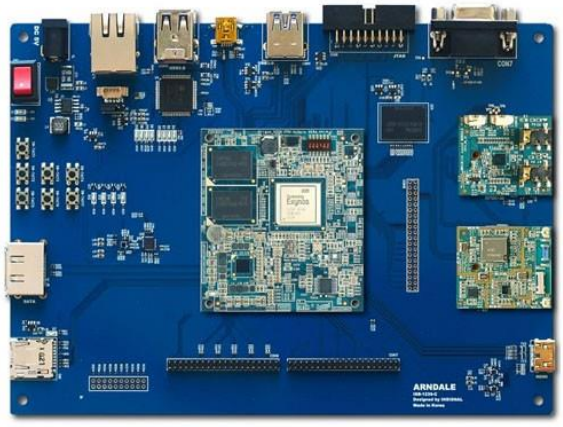


Figure 2.7.2-1 The Exynos 5-based Arndale community board for application developers.

[9] http://www.nvidia.com/object/tegra-x1-processor.html

[10] http://www.engadget.com/2015/01/04/nvidia-tegra-x1/

[11] http://www.arndaleboard.org/wiki/index.php/File:5420_block.jpg

4.2 General Purpose Processors

The interfacing of a modern imager is a very power consumption demanding procedure enabling hardware for deserialization and ISP (see Chapter 2.4). CPU manufacturers have integrate into the processors IP core the necessary logic to interface with imagers. Intel has integrate into mobile processors the necessary logic to enable deserialization and make ISP functions. The result is a processor capable of interfacing with three image sensors simultaneously (figure 4.2). The main disadvantage of this design (regarding a reconfigurable imaging system) is the embedded API that enables only specific controls over imagers. This approach is more common because of the ease of use by application developers, which are the targeted market for this product/feature. Moreover the supported imagers with MIPI CSI-2 as mentioned above are limited and in with mid-range performance.

One aspect of this solution that is unclear is the way the imager is connected to the CPU and some technical details regarding the other necessary electronics for power supply. The imagers tend to have a wide range of voltages in the LVCMOS vicinity (from 1.2 to 2 Volts) for power supply and in many cases they demand 2 or 3 different voltages in order to function properly [4.2].

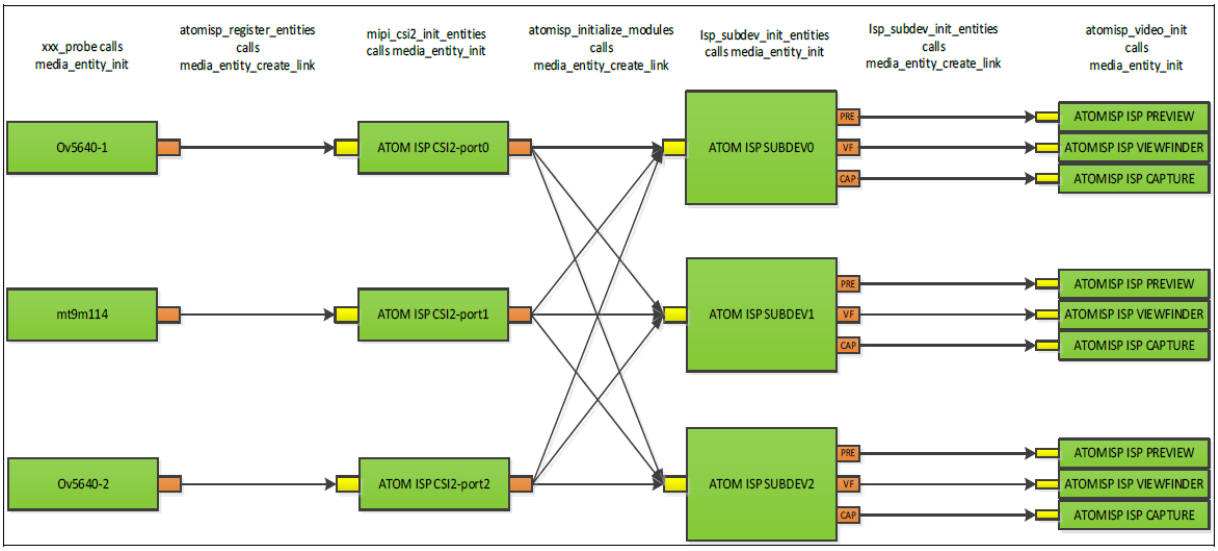


Figure 2.7.2-1 The Intel ATOM CPU interconnection with 3 different imaging sensors.

4.3 Field Programmable Gate Arrays – FPGAs

FPGAs was the alternative solution when other hardware couldn't get the job done. The flexible design can focus on the important part and leave the secondary task for later. Many examples regarding FPGAs are presented to back up this theory: [4.3, 4.4, 4.5, 4.6, 4.7, 4.8]. As reviewed previously modern imagers need differential signaling in order to transfer serial data. Most FPGAs feature differential signaling in many voltage levels. Many of them have a variable voltage I/O which can change from subLVDS to LVDS voltage levels [9]. The FPGA is the safest solution in terms of functionality. After a quick reading of the articles mentioned earlier, one can make the assumption that an IP core design for a system as complex as an imaging system is not an easy task. The IP core design need to be done from scratch without any reference designs. Many components essential for the frame acquisition and transfer like Serializers/Deserializers – SerDes pre-exist as "black boxes" in other hardware. In FPGA design they need to be constructed and tested, which takes a lot of time. One last drawback of FPGAs is the fact that although been expensive as hardware, they have more expensive tools. A tester for hardware implementation costs nearly as the FPGA itself!

To conclude with, the FPGA scheme is undeniably a powerful tool. Functions like the deserialization of serial data is the reason to use them. As mentioned at Chapter 2.4 manufacturers like Lattice have developed FPGAs to serve as bridges between sensor and application processor. This way the need for extended IP core design is eliminated as the selected FPGAs serve a specific cause, leaving the rest of the work to dedicated hardware.

Figure 2.7.2-1 Xilinx Vertex 6 FPGA with a PCI-X communication for fast image data transmission.

4.4 Data Acquisition – DAQ Cards

This approach has not been presented and is an experimental concept for the scope of this thesis. The main idea is the total transfer of the sensors signals to the host PC using a DAQ card in order to be processed and extract the image information. This enables total control over the image sensor without extra physical layers and delays.

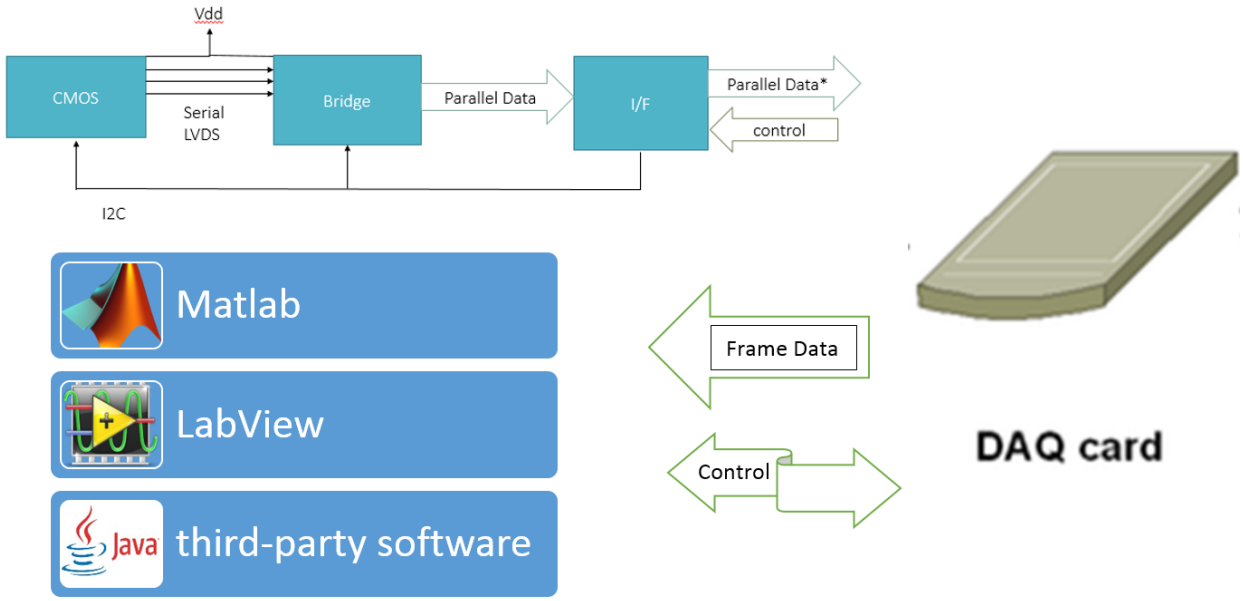


Figure 2.7.2-1 The Frame grabber featuring a DAQ card and a bridge to desirialize the data and make them readable to the card.

In figure 4.4 it is shown how the final system will be connected. First a bridge like the one from Lattice will make the data "slow" enough for the DAQ card to read by desirialize them. Next the DAQ card will sample the signal values and write them in a buffer for the final application to read.

The signals coming from the sensor are Digital, but a DAQ card treads them like analog. According to Nyquist theorem the sampling rate must be at least double in order to prevent aliasing and loss of information. Adding 4 MHz for the Jitter effect [4.11] the minimum sampling rate is calculated next:

10 – 16 bit: parallel pixel Data -> out

1 bit: pixel_clk -> out

1 bit: Frame_Valid -> out @ sensor dep avg , 100MHz max

1 bit: Line_Valid -> out @ sensor dep avg , 100MHz max

1 bit: Global reset -> in

3 bit: I2C/3 wire serial -> bidirectional @ 100MHz max

2 X Max camera Data Rate + c = 420.928MHz

Where c = 4 MHz because of the Jitter effect.

- ⇨ Read out time: 2.37 nsec
- ⇨ DAQ cards with I/O sampling rate @ 420.928MHz
  Data Rate @ 208.464Mbps if Digital acquisition is supporded.
- ⇨ Fast Cards with 24 Channels have Real-time hardware comparison at up to 200 MHz
- ⇨ Because of the fast readout the use of Dynamic (Clocked) Acquisition is mandatory.

Unfortunately the only products with the required specifications available was from National Instruments.

NI PXIe-6544 (Data read rate: 400 Mbits/s ,32 channels)

NI PXIe-6548 (Data read rate: 300 Mbits/s ,32 channels)

Datasheet : NI PXIe-6544, PXIe-6545, PXIe-6547 and NI PXIe-6548

NI PXIe-6555 (Data read rate: 200 Mbits/s , 32 channels)

Datasheet : NI PXIe-6555, NI PXIe-6556

NI PCIe-6536B (Data read rate: 200 MB/s maximum throughput, 32 channels)

Datasheet: NI PCIe 6535B, 6536B, 6537B

NI 6587R (data read : 1 Gbit/s LVDS on 20 channels (use without bridge, straight to cmos)).

Datasheet

This solution was discarded finally due to the extremely high cost of the cards (>7500 GBP). This was expected in some level because of the technology and performance behind them.

4.5 Serial Interface Frame Grabbers

The use of frame grabber cards for scientific imaging is known for many years. They use serial protocols like CXP and HSLink [4.12, 4.13]. The advantages of using serial protocols like CXP and HSLink are numerus. More important is the bandwidth they offer in just a simple cable. The bit rate can reach 6.25Gbps transmission and receive data at 20Mbitps. This is called asymmetric high speed point to point serial communication for the transmission of video and still images, scalable over single or multiple coaxial cables. One more thing they have to offer is the ability to connect long cables (up to 200 m) and use any cable length. This is possible because of an equalizer chip making adjustments to the serial signals.

Both protocols only interface with camera systems and there is no way to connect an image sensor directly to them. The approach to this solution is based on the development of the chip system, able to interface with the various imagers. The most common chip for CXP comes from EqcoLogic (figure4.5). These transceiver chips are connected to the camera electronics and transmit the frame data while receiving control uploads from the host. For HSLink an additional serializer is required [4.14] raising this solutions complexity.

The complexity of the chip integration became an issue when the technical details came into consideration [4.15].

On the other hand frame grabber manufacturers were more than helpful. The most important downside was the price tag reaching approximately 1000 euros[12].

The use of frame grabbers as the main platform under the burdened of the severe disadvantages was quickly abandoned. In future researches it is advised to take into consideration this solution the above issues are solved with new hardware.

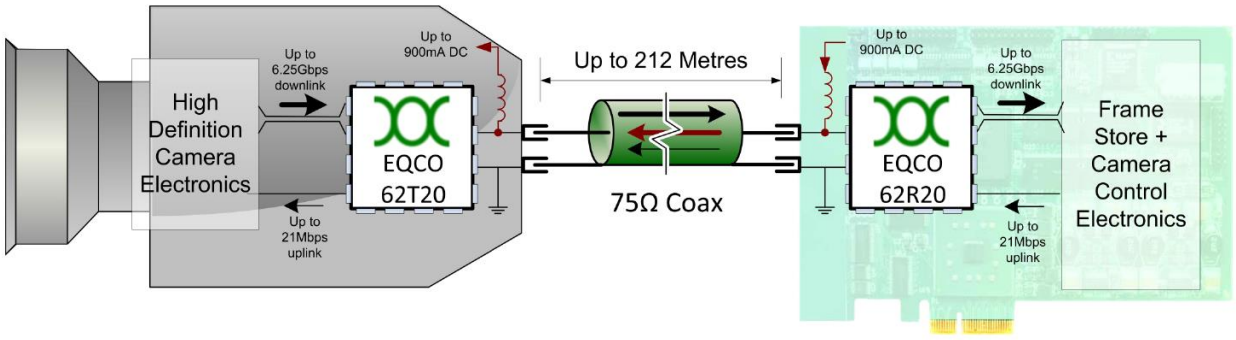


Figure 2.7.2-1 The complete frame grabber solution.

[12] http://www.matrox.com/imaging/en/products/frame_grabbers/

4.6 Embedded Video Interfaces

Specialized microcontrollers and ASIC chips are great assets for complex designs as they simplify the system parts while delivering in many cases the best performance possible. The struggle for performance in many electronic imaging systems can be compensated by the use of this kind of hardware.

An embedded video interface is the existence of a preconfigured hardware to communicate with a host and delivering video frames via connection protocols like USB3 Vision, Gig Ethernet and fire wire[13]. There is one basic requirement: The protocol been utilized must have the necessary bandwidth to transfer high definition frames at high frame rates. Real world measurement shows that the effective bandwidth available via the USB3 bulk transfer in method is around 400 MByte/s; approximately 10 times that of USB 2.0 [4.16].

After reviewing a few choices, one made an impression. The CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress is an inexpensive ($49.00) and well documented solution offering USB 3.0 connectivity. This is the result of the FX3 specialized microcontroller featured by many other board like the one above, making this design a future proof choose for the development of an extended electronic imaging platform. The main advantage of this board is that it has the capability to interface directly with a parallel data imager and an FPGA bridge like the Lattice bridge or use other FPGAs interconnection like Xilinx FPGA boards and Altera FPGA boards. The FX3 controller in combination with the Lattice bridge is available by Lattice itself in a more expensive package. The advantages of this board over the SuperSpeed Explorer Kit from Cypress is that it combines the advantages of FPGAs and embedded video interfaces to interface with imagers using serialize protocols like subLVDS and MIPI CSI-2.

An even more agile solution comes from Net Vision featuring a “sensor tester” board reconfigurable to many voltage levels of power supply or I/O[14]. Unfortunately the manufacturer informed me that is not possible to ship the product from Japan to Greece. An alternative solution was then proposed by the manufacturer involving the cooperation with a University in Japan in order to acquire the product. In a future research it is advised to take into consideration the acquirement of this hardware.

[13] http://www.pleora.com/support-center/documentation-and-downloads/iport-ntx-u3-embedded-video-interface-data-sheet

[14] http://www.net-vision.co.jp/sv-english/svm-03U.html

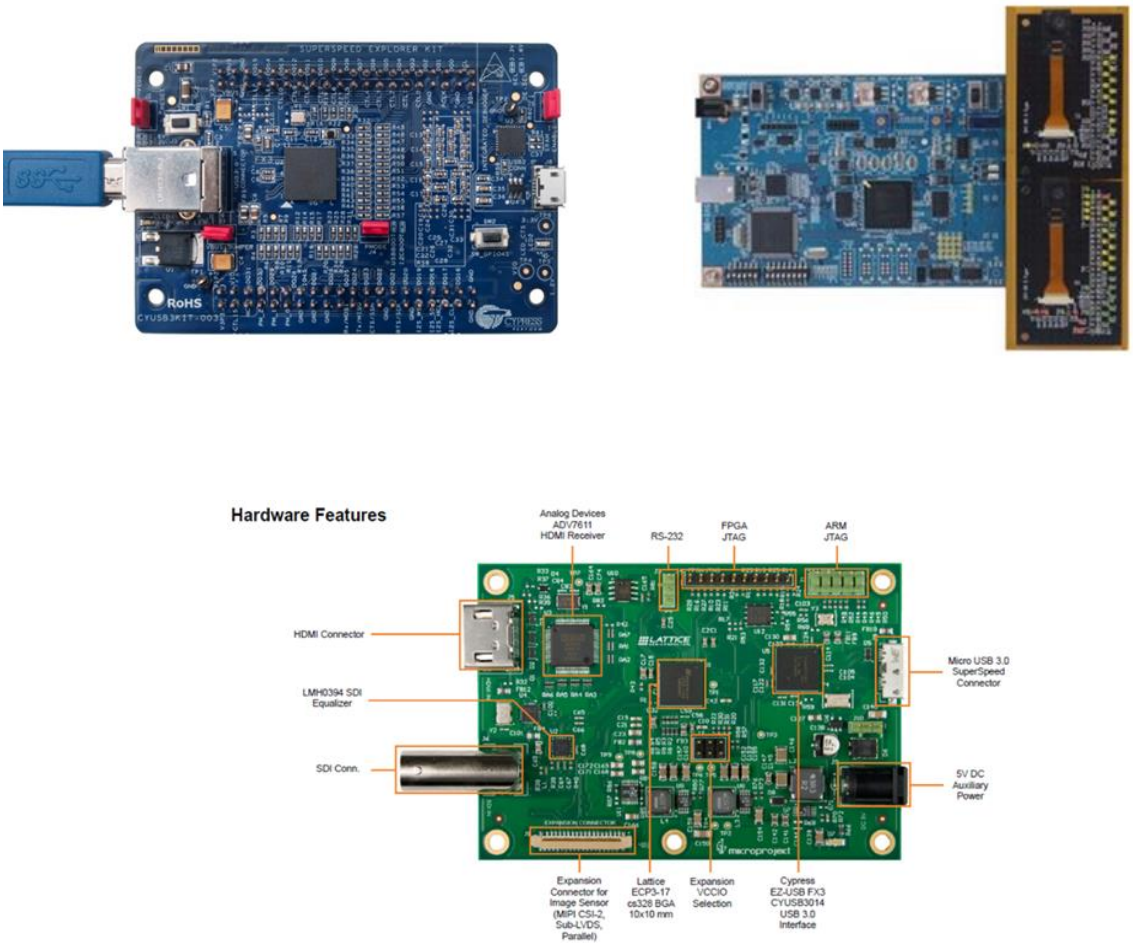


Figure 2.7.2-1 The three boards found featuring the FX3 USB3 controller: On the top left the CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress, next the Net Vision SVI-06 and below the Lattice USB 3 Board , all offering great platform reconfigurability and hardware performance.

4.7 Conclusion

After taking into consideration all the advantages and disadvantages of each solution proposed, the results are in favor of the embedded video interface approach. More precisely a design involving the CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress will be developed by this thesis, and another one involving the Lattice USB 3 Board will be proposed as future work to expand platform reconfiqurability to new imagers featuring serial data transfer.

5 PLATFORM IMPLEMENTATION

In this chapter the platform development and implementation will be explained in detail. The design approach may refer many times to previous references in chapter 2 consisting the theoretical Background around a specific topic.

5.1 EZ-USB FX3 Controller Architecture

The EZ-USB FX3 controller is a general purpose integrated USB 3.0 SuperSpeed controller with a built-in programmable interface (GPIF™ - II) and support for accessing a set of serial peripherals. The peripherals and the GPIF - II will be examined latter in more detail.

This controller provides easy connectivity to popular interfaces, such as asynchronous SRAM, asynchronous and synchronous address data multiplexed interfaces, and parallel ATA. FX3S has integrated the USB 3.0 and USB 2.0 physical layers (PHYs) along with a 32-bit ARM926EJ-S microprocessor for powerful data processing and for building custom applications. It implements an architecture that enables 185-MBps data transfer from GPIF II to the USB interface.

The main future provided by the EZ-USB FX3 controller is the ability to interface with an application processor and in our case an image sensor (see figure 5.1-1).

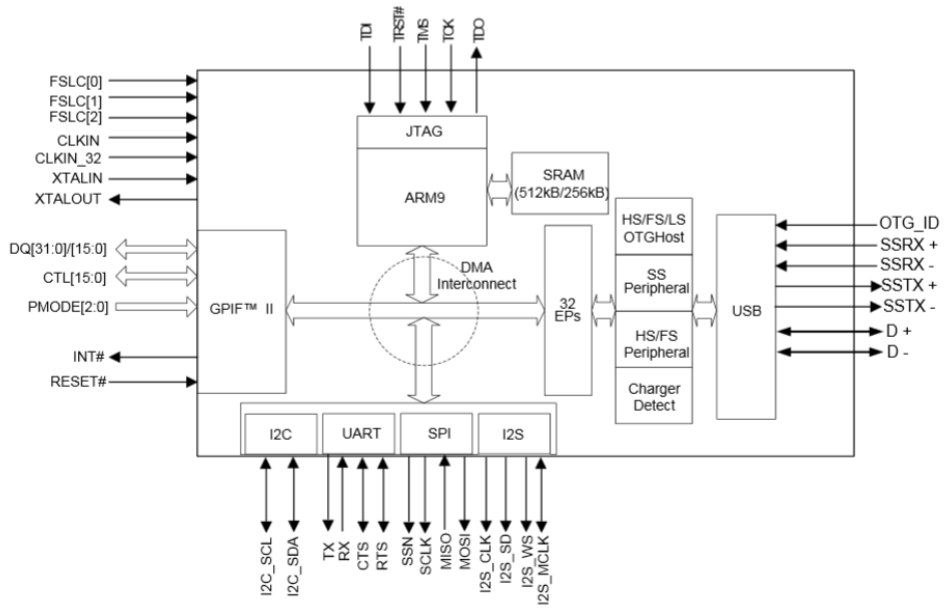


Figure 2.7.2-1 The functional diagram of the EZ-USB FX3 controller.

The FX3 utilize a 32-bit ARM926EJ-S ARM9 microprocessor with 512 kB RAM CPU [1]. The ARM926EJ-S is suitable for multi-tasking applications where full memory management, high performance, low die size, and low power are important. It supports both the 32-bit ARM and 16-bit Thumb instruction sets, enabling you to trade-off between high performance and high code density. Cypress provided Application Program Interface – API can run optimal with the provided compiler based on Eclipse. The core has direct access to 16 KB of instruction tightly coupled memory (TCM) and 8 KB of data TCM. The ARM926EJ-S core provides a JTAG interface for firmware debugging. As it is illustrated by Figure 5.1-2, ARM926EJ-S has a Harvard cached architecture providing a complete high-performance processor subsystem that includes:

- an ARM9EJ-S integer core
- a Memory Management Unit (MMU)
- separate instruction and data AMBA AHB bus interfaces
- Separate instruction and data Thirdly Coupled Memories (TCM) interfaces.

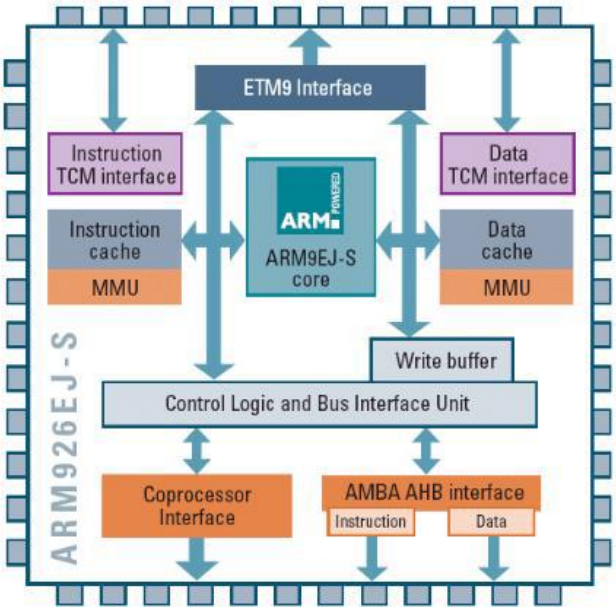


Figure 2.7.2-2 ARM926EJ-S Block Diagram

FX3 integrates 512 KB or 256 KB of embedded SRAM (depending on the part number selected) and supports four methods for booting the code: USB, GPIF II, I2C, or SP. The selected method can be choose by a provided software called Control Center in combination with hardware overwrite with jumpers.

At last FX3 enables efficient and flexible DMA connectivity between its various peripherals (such as, USB, GPIF II, I2S, SPI, and UART). The DMA used in this application is between the USB and the GPIF II. After the FX3 firmware configures data accesses between peripherals, the DMA fabric manages transfers without involving the ARM9 core. In this application the ARM9 core will be notified when the frame transfer ends to add a header and then relay the information to the USB socket. This implementation is called a manual DMA channel and it will be discussed in more detail in the next topic.

5.1.1 GPIF – II Peripheral

The high performance GPIF II peripheral (a part of the processor interface block (PIB) featured by the FX3 controller) enables functionality involving the interface with any application processor including an image sensor. GPIF II is a programmable state machine that enables a flexible interface running on its own high speed clock, autonomous to the ARM9 core. GPIF II may either function as a master or slave in industry standard or proprietary interfaces. Both parallel and serial interfaces may be implemented with GPIF II. The interface used in this design is the parallel synchronous interface with the GPIF II state machine functioning with external clock.

The key features of GPIF II are:

- Functions as master or slave.
- Provides 256 programmable states.
- Supports 8 bit, 16 bit, 24 bit, and 32 bit parallel data bus.
- Supports interface frequencies up to 100 MHz (74.25 MHz used in this design).
- Supports 14 configurable I/O pins (to function as control signals) when a 32 bit data bus is used. Control pins can be input, output, or bidirectional.
- Supports 16 control I/O pins when a 16/8 data bus is used. Control pins can be input, output, or bidirectional.

Cypress's GPIF II Designer Tool enables fast development of GPIF II state machines. The GPIF II Designer Tool is available with the EZ-USB FX3 SDK installation provided by Cypress. The interface selection and options are shown in figure 5.1.1 below. More details about the GPIF II Designer Tool will be discussed in the firmware implementation.

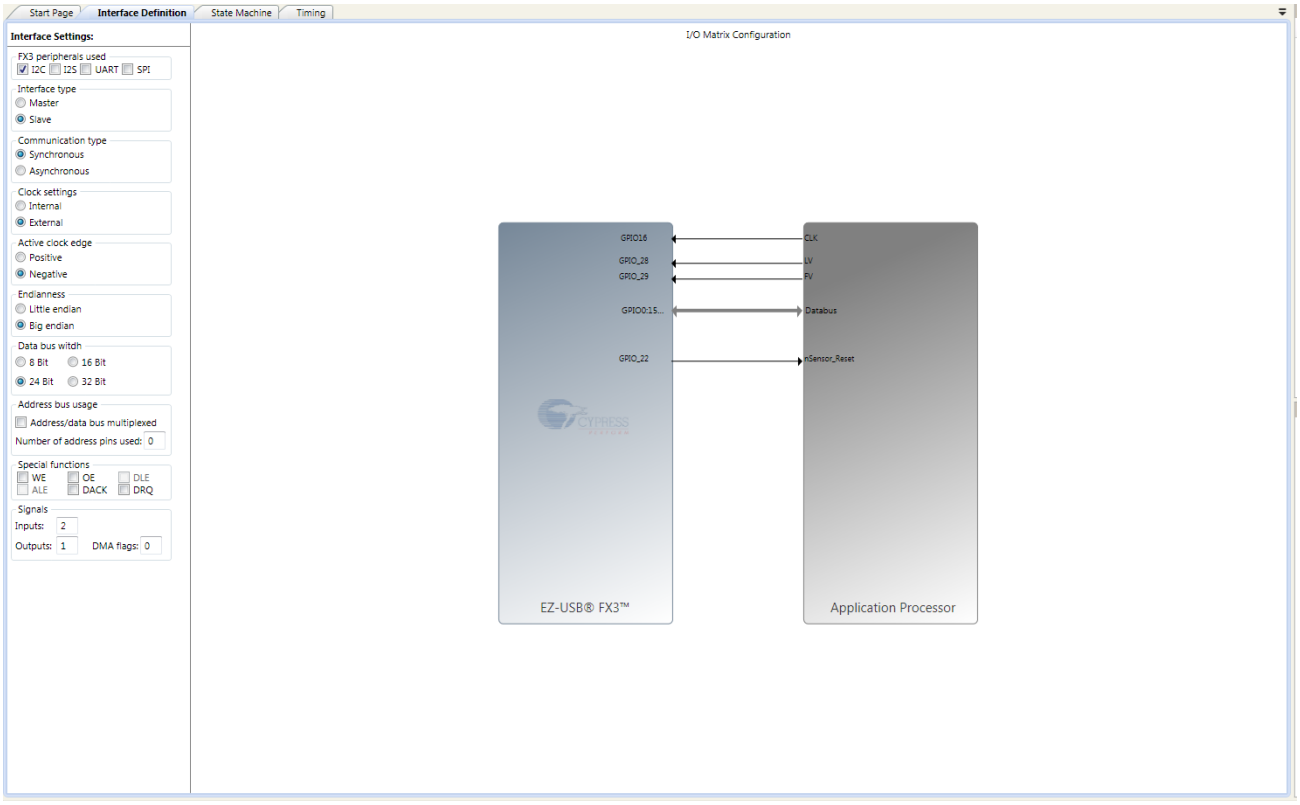


Figure 5.1.1-1 The GPIF II Designer Tool enables the interface parameter selection resulting in an easy and fast implementation of the interface data with, communication blocks utilized and specific control characteristics.

5.1.2 I^2C Interface

I^2C (Inter-Integrated Circuit) is a multimaster serial single-ended bus used for attaching low-speed peripherals to an electronic device such as a motherboard, an embedded system or a smartphone.

FX3's I^2C interface operates as an I^2C master enabling communication with I^2C slave devices. For example, FX3 may boot from an EEPROM connected to the I^2C interface. FX3's I^2C master controller supports multi master functionality and allows for I^2C clock stretch. The bus frequencies supported by the I^2C controller are 100 kHz, 400 kHz, and 1 MHz. When the power domain of I^2C is 1.2 V, the maximum operating frequency supported is 100 kHz. When the power domain of I^2C is 1.8 V, 2.5 V, or 3.3 V, the operating frequencies supported are 400 kHz and 1 MHz.

FX3's I2C interface can operate only as I^2C master.

Both SCL and SDA signals of the I2C interface require external pull-up resistors, which must be connected to I2C power domain.

From the GPIF II Designer Tool the I^2C block can be used for the image sensor communication. The sensor receives and transmits data thru I^2C making possible the sensors power up and control.

5.1.3 JTAG Interface

JTAG is the common name for the Standard Test Access Port and Boundary-Scan Architecture. JTAG is used as the primary means of accessing sub-blocks of integrated circuits, making it an essential mechanism for debugging embedded systems which may not have any other debug-capable communications channel.

FX3's JTAG interface has a standard five-pin interface to connect to a JTAG debugger to debug firmware through the CPU-core's on chip debug circuitry. Industry standard debugging tools for the ARM926EJ-S core can be used for the FX3 application development.

Although the JTAG interface is not been utilized in this design, it can be used to debug the Lattice USB 3 board featuring the FX3 controller in future platform development.

5.1.4 UART Interface and Integrated Debugger

UART stands for universal asynchronous receiver/transmitter and is a computer hardware device that translates data between parallel and serial forms. The UART interface provides asynchronous serial communication with other UART devices operating at speeds of up to 3 Mbps. It supports 7 to 8 data bits, 1 to 2 stop bits, odd, even, mark, space, and no parity. The UART interface supports full duplex communication with a signaling format that is compatible with the standard UART protocol. FX3's UART interface consists of the TX, RX, CTS, and RTS signals. The UART is capable of generating a range of baud rates, from 300 bps to 4608 Kbps, selectable by the FX3 firmware.

Finally the UART is used by the integrated debugger: The CY7C65215 USB-Serial IC. The CY7C65215 USB-Serial IC is a full-speed USB-Serial bridge controller that offers two configurable serial communications blocks (SCBs). The development board uses the first serial channel of the CY7C65215 as a UART and the second serial channel as a JTAG interface. Figure 5.1.4 shows the block diagram [2].

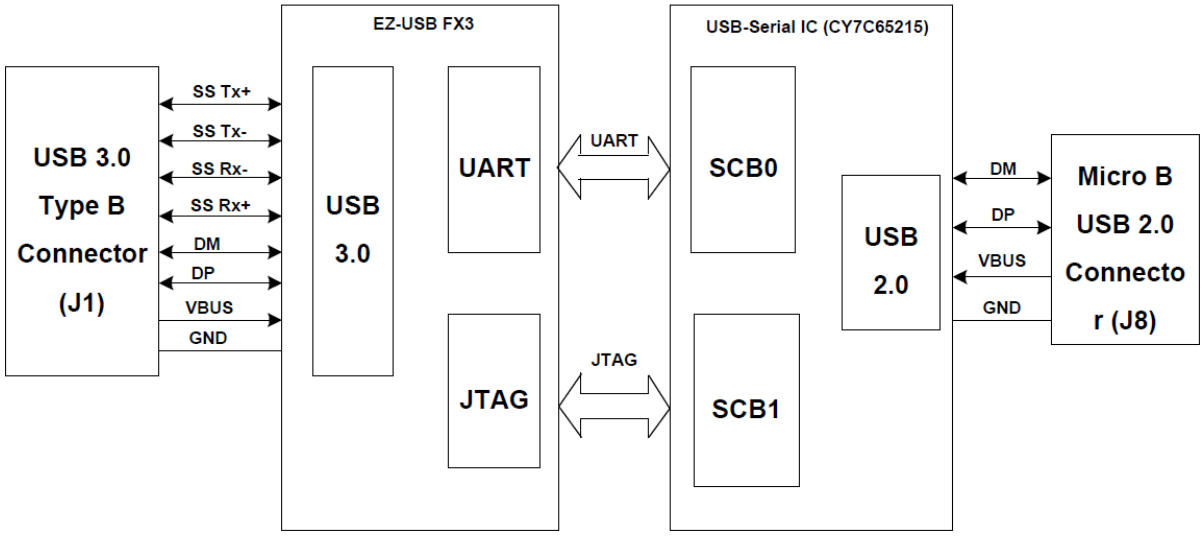


Figure 5.1.4-1 Block Diagram of CY7C65215.

After USB Serial IC Enumerate as COM Port in the host PC a PC terminal application can be used for listening debugging massages from firmware to the bound rate indicated. Tera Term application was used without any problems in this design.

5.2 Hardware Implementation

In this topic the CMOS image sensor and the custom Printed Circuit Board – PCB featured in the CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress will be discussed.

5.2.1 MT9M034 CMOS Image Sensor

The MT9M034 CMOS digital image sensor from Aptina (now known as ON Semiconductor)[] utilize a 1/3 inch area with an active-pixel array of 1280Hx960V. It features an Electronic Rolling Shutter – ERS which makes good image quality sustainable only if high frame rates are used (see Chapter 2.3.2 for ERS artifacts). Fortunately one of the technical characteristics is the 60 fps for 720p resolution, which is enough to eliminate most ERS artifacts for moving objects. The functional block diagram is shown in figure 5.2.1-1.

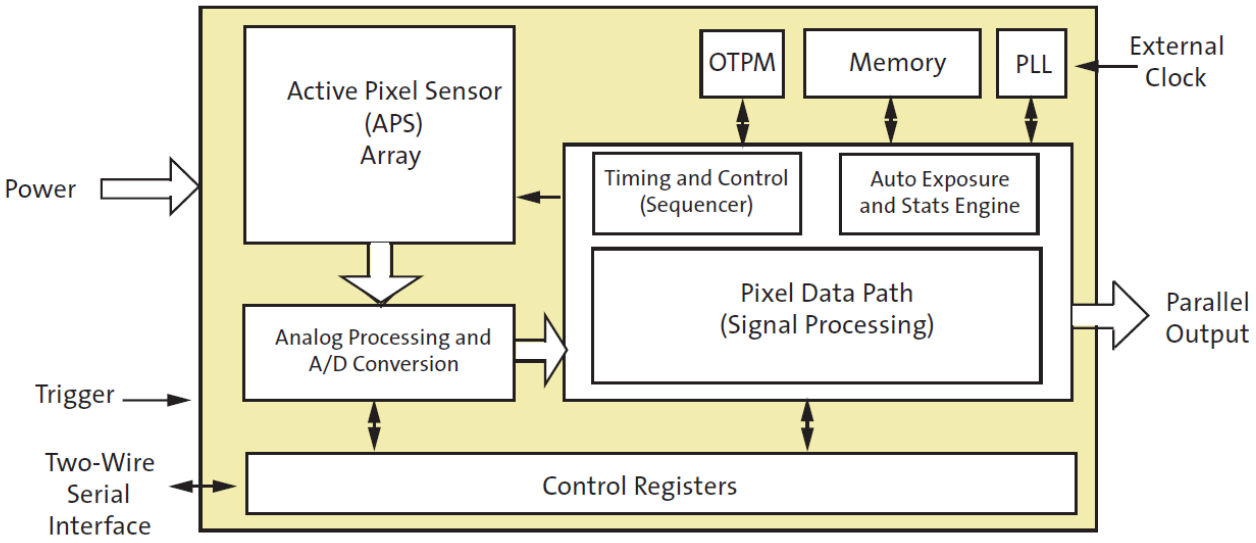


Figure 5.2.1-1 Block Diagram of MT9M034 CMOS image sensor.

As shown by figure 5.2.1-1, the sensor receives an external clock which in this design is at 74.25 MHz. The Trigger pin for still images is not used for this design. The two-wire serial interface (I^2 C) is connected with the board in a Master configuration. This connection allows the proper setup of the sensor parameters and enables the total control of the function in an "on the fly" application like the human vision emulation where the use of context switch changes the number and area of pixels that the sensor is going to read (Region Of Interest – ROI).

The low light performance of the MT9M034 is excellent with QE that reaches 63% at 540nm wavelength. Note that this kind of performance is expected by a FSI sensor (see Chapter 2.3.3). A better performing BSI sensor would have QE reaching 70 or as high as 85%. The CFA plays catalytic role to QE as it absorbs part of the incident radiation. This model is a color image sensor meaning it utilize a CFA covering the pixel array. The CFA is a "grbg" Bayer structure meaning even-numbered rows contain green and red pixels, odd-numbered rows contain blue and green pixels. The first physical active pixel for readout starts from the right of the pixel array as shown in figure 5.1.2-2.

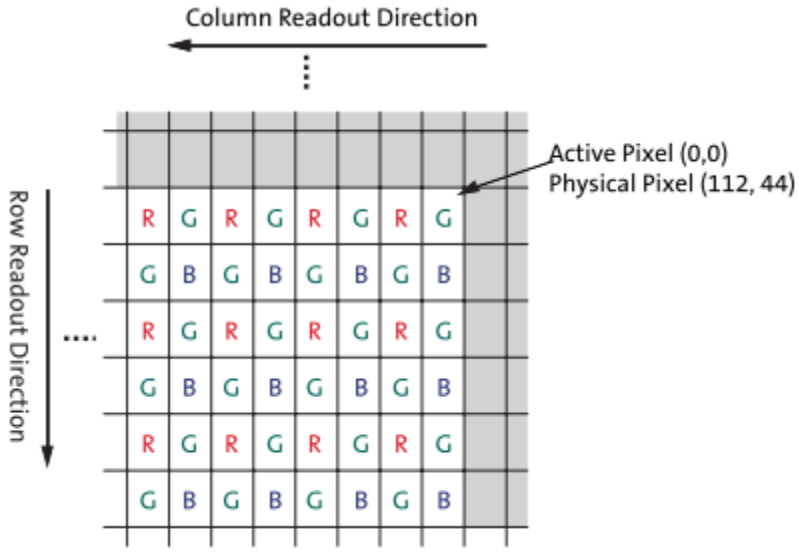


Figure 5.2.1-2 Color filter pattern (Bayern pattern) of sensor's active pixel array.

Context Switch is the concept of changing the active registers containing image parameters between two sets: Context A and context B. In each frame the active registers set can change resulting to different image on the next frame [5.6].
Once a row has been read, the data produced from the columns is sequenced through an analog signal chain (providing offset correction and gain) and then through an analog-to-digital converter (ADC). The output from the ADC is a 12 bit-value for each pixel in the

array. The last stage before the sensor delivers the actual 12 bit RAW data is a digital processing signal chain, which provides further data path corrections and applies digital gain.
The frame readout was introduced in Chapter 2.4. Here in figure 5.2.1-3 we see a spatial representation of the Horizontal and Vertical Blanking resulting when the concord signals are inactive.

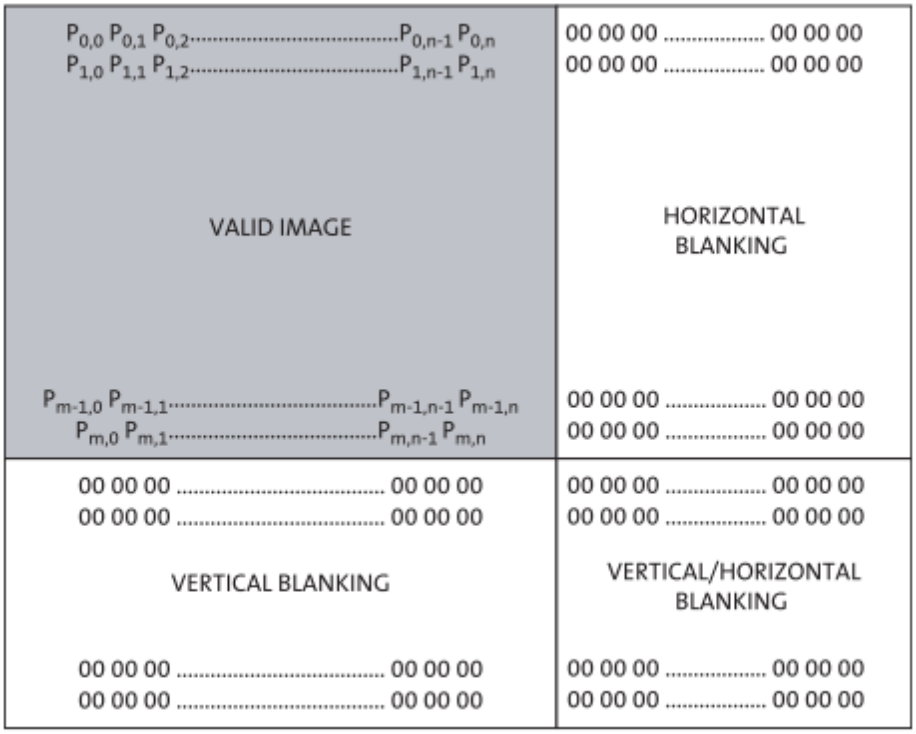


Figure 5.2.1-3 Spatial representation of image readout.

The PCB containing necessary hardware for the sensor function also known as sensor front end is part of the product manufactured by Leopard imaging (figure 5.2.1-4).

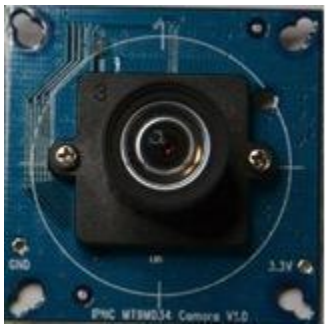


Figure 5.2.1-4 Front view of the LI-CAM-M034 camera board with a 6mm lens mounted.

5.2.2 Image Sensor Board

The CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit from Cypress needs to physically connect with the application processor or in our case the image sensor. The manufacturer of the LI-CAM-M034 camera board used a molex adaptor [5.7] (part no 525593652, figure 5.2.2-1) for the camera board interconnections. In order to connect the sensor camera board to the GPIF II pins onboard the CYUSB3KIT-003 a custom Printed Circuit Board – PCB is required.

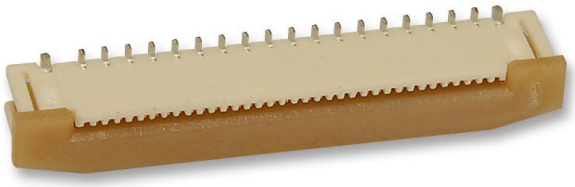


Figure 5.2.2-1 The molex FFC connector used for the PCB fabrication.

Then the PCB was designed, printed and tested for connectivity and structural errors. The errors found were minimal and fixable. The biggest issue was the fine structural detail required for the molex connector, causing some structural difficulties.
For the right connections to be made, the GPIF – II manual was studied in order to meet the manufacturer's requirements [5.8]. According to the schematics, the connections between the CYUSB3KIT-003 EZ-USB® FX3™ SuperSpeed Explorer Kit and the LI-CAM-M034 camera board are shown in figure 5.2.2-2 below.

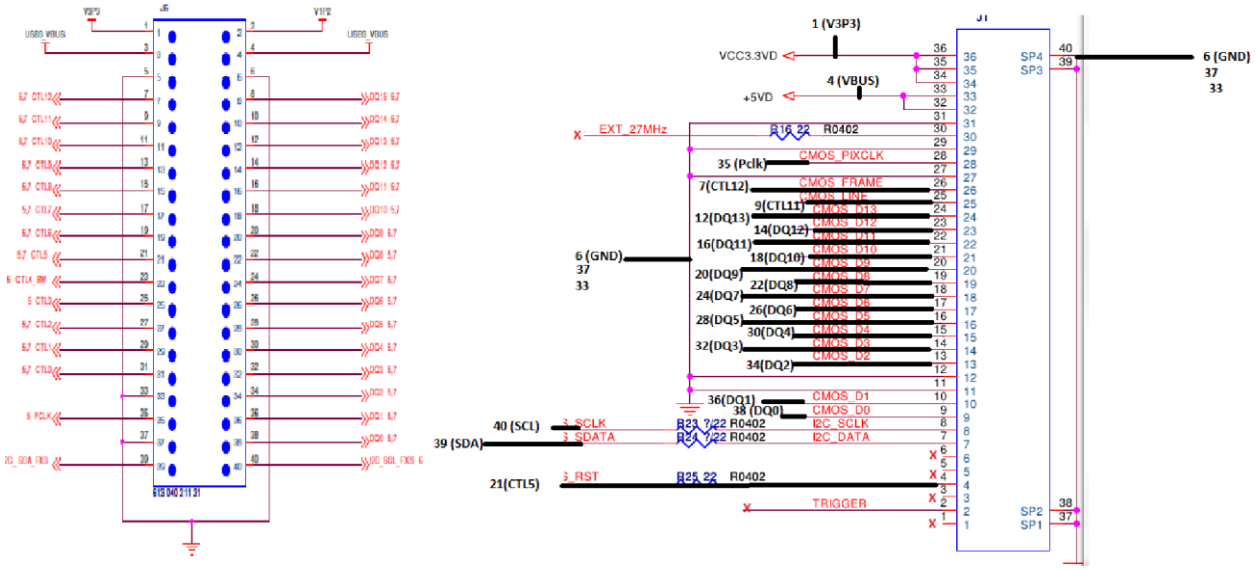


Figure 5.2.2-2 The necessary connection between the controller board and camera board.

Next the PCB, camera board and controller board were able to be connected as shown by figure 5.2.2-3.

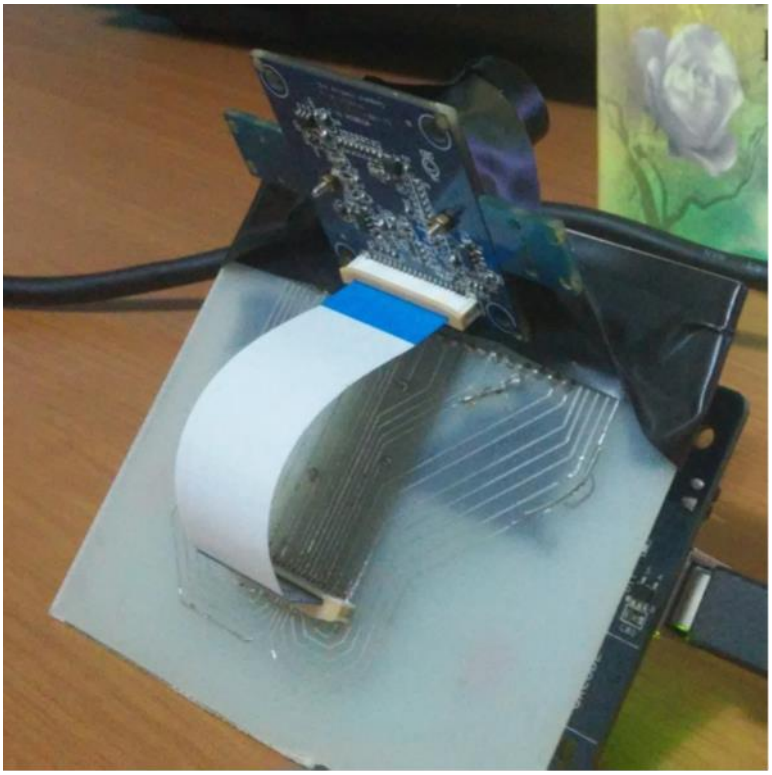
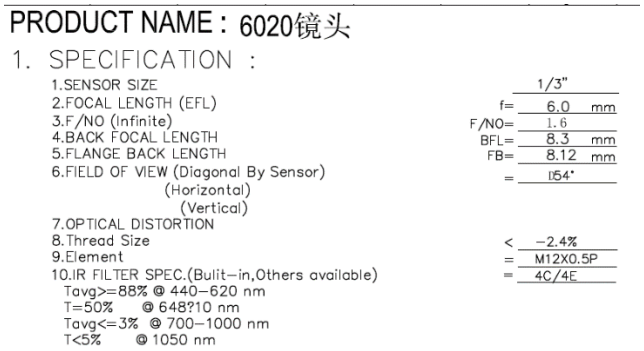


Figure 5.2.2-3 The final hardware connected and fully functional.

5.2.3 Image Sensor Board Optics

The knowledge of the optics specifications and technical characteristics are vital information in order to make the right calculations in the System Spatial Resolution presented by Chapter 6.2. The camera board manufacturer provided the technical specifications for the camera board and the following characteristics are extracted:



PRODUCT NAME : 6020镜头

1. SPECIFICATION :

1.SENSOR SIZE	1/3"
2.FOCAL LENGTH (EFL)	f= 6.0 mm
3.F/NO (Infinite)	F/NO= 1.6
4.BACK FOCAL LENGTH	BFL= 8.3 mm
5.FLANGE BACK LENGTH	FB= 8.12 mm
6.FIELD OF VIEW (Diagonal By Sensor)	= 154°
(Horizontal)	
(Vertical)	
7.OPTICAL DISTORTION	
8.Thread Size	< −2.4%
9.Element	= M12X0.5P
10.IR FILTER SPEC.(Bulit−in,Others available)	= 4C/4E

Tavg>=88% @ 440−620 nm
T=50% @ 648?10 nm
Tavg<=3% @ 700−1000 nm
T<5% @ 1050 nm

5.3 FX3 Firmware

Cypress provides software and firmware stack for FX3 to integrate all USB applications in the embedded system environment (in our case an Embedded Video Interface). The software development kit comes with application examples that accelerate application development. The firmware developed in this design is based on the application note AN75779 - How to Implement an Image Sensor Interface with EZ-USB® FX3™ in a USB Video Class (UVC) Framework [5.3]. The changes and additions made are discussed in this topic along with the applications notes. Next the Firmware API and Firmware Stack features will be discussed.

Firmware Stack: The Firmware stack provided by Cypress enables the build of powerful and flexible applications rapidly using FX3 firmware framework and FX3 API libraries (figure 5.3-1). The API libraries are:

- Register count
- Parallel data paths: FX3 allows parallel data paths between the various peripherals and a much more complicated DMA mechanism.
- Modular approach: An API-based approach means that the customer ends up writing minimal code. This approach is easy to use, debug, modify, and support.

Firmware API: The API library and the corresponding header files provide all the APIs required for programming the different blocks of the FX3. The APIs can be classified as high level and advanced.

- High-Level APIs: These include APIs for USB, GPIF II, DMA, serial interfaces (UART, I2C, I2S, SPI, GPIO), and OS system call functionality.

- Advanced APIs: These include APIs for debug, power management, low-level DMA blocks, and advanced OS system call functionality.

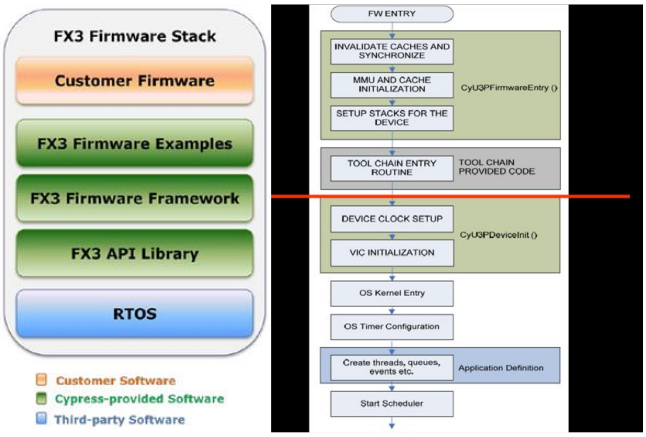


Figure 5.2.3-1 The firmware stack on the left and the Firmware Sequence on the right.

The firmware use Standard UVC drivers that are available in most operating systems, including Windows, Linux, and Mac. They do not require a vendor-specific driver. This way the platform can be "plug and play" with only requirement the use of the custom application to modify the received pixel values and create the final RGB image. One important requirement of the platform is the use of a USB3 port in order to exploit the full potential bandwidth.

Although the system is running without any custom drivers FX3's firmware needs to provide UVC class-specific descriptors to the PC host in order to enumerate properly. FX3 is responsible for the following:

- Setting up USB descriptors for enumeration
- Configuring the image sensor
- Handling UVC host commands
- Adding UVC header to raw video data before passing it to the USB 3.0 host

The operating system implemented by the manufacturer to run by the ARM9 core is called ThreadX. ThreadX is a Real Time Operating System – RTOS which means that is characterized by predictable time response and deterministic behavior. These properties are attributed to features such as sophisticated scheduling, minimal interrupt latency and minimal thread switching latency. Developing on a RTOS enables you to conveniently design a system with deterministic behavior; not that without it you are not able to do so.

The development is done in Eclipse Integrated Development Environment – IDE:

The Eclipse IDE for C/C++ Developer is provided as part of the FX3 SDK. This IDE comprises the base Eclipse platform and the CPP feature. A couple of plug-ins required for development are bundled with the IDE.

- GNU ARM C/C++ Development support
- Zylin Embedded CDT
- Java(TM) Platform, Standard Edition Runtime Environment Version 7 (JRE)

The use of a USB 3.0 Protocol Analyzer is not necessary because of the standard USB Video Class implementation. If a custom class needed to be implemented then a USB 3.0 Protocol Analyzer can capture all of the 3.0 bus packet transfers between the host and device and decode it into easily understandable software so that you know the USB transfer packet on the bus. The USB 3.0 protocol analyzer reduces the design and debug time in 3.0 applications.

5.3.1 USB Video Class – UVC

As described above the device when connected to the USB 3.0 port needs to provide the necessary data in order to enumerate properly. A file, named cyfxuvcdscr.c that contains the UVC enumeration data. The USB specification, which defines the format for UVC Descriptors, can be found at usb.org [4]. The following is a high-level view of the Descriptors. A UVC device has four logical elements (see figure 5.3.1-1):

1. Input Camera Terminal (IT)
2. Output Terminal (OT)
3. Processing Unit (PU)
4. Extension Unit (EU)

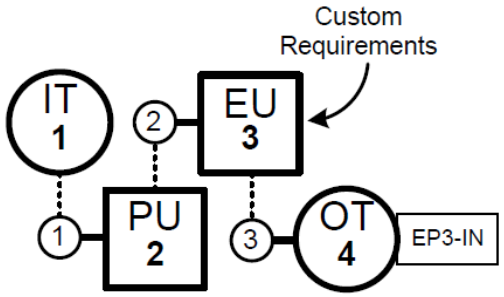


Figure 5.3.1-1 UVC Diagram of the Camera Architecture

Connections are made between elements by associating terminal numbers in the Descriptors. For example, the Input (Camera) Terminal Descriptor declares its ID to be 1, and the Processing Unit Descriptor specifies its input connection to have the ID of 1, logically connecting it to the Input Terminal. The Output Terminal Descriptor specifies which USB Endpoint to use, in this case BULK-IN Endpoint 3. The Descriptors also include video properties, such as width, height, frame

rate, frame size, and bit depth, and control properties, such as brightness, exposure, gain and contrast.

After the Host enumerates the camera, the UVC driver sends a series of requests to the camera to determine operational characteristics. This is called the capability request phase. It precedes the streaming phase, in which the Host application starts streaming video. The FX3 firmware responds to the requests that arrive over the USB control Endpoint (EP0).

When a Host application makes a request to change the brightness value, the UVC driver issues a SET control request to change the brightness value (SET_CUR). Then the variable connected to the request passes thru a check to be converted as a valid parameter for the image sensor. For example the brightness of 55 (for a range from 0 to 255) is the integration time that the imager collects photons. The parameter is multiplied by a constant to meet the real time needed for this brightness value.

5.3.2 Video Data Format: YUY2 and RGB24

The details around YUY2 and RGB24 were discussed previously in Chapter 2.6.1 for YUY2 and in Chapter 2.6.2 for the RGB24. Here a new issue will be discussed involving the UVC Video Data Header needed in order to send the frames properly thru UVC. The UVC class requires a 12-byte header for uncompressed video payloads. The header describes the properties of the image data being transferred. For example, it contains a "new frame" bit that the image sensor controller (FX3) toggles for every frame. The FX3 code also can set an error bit in the header to indicate a problem in streaming the current frame. This UVC data header is required for every USB transfer.

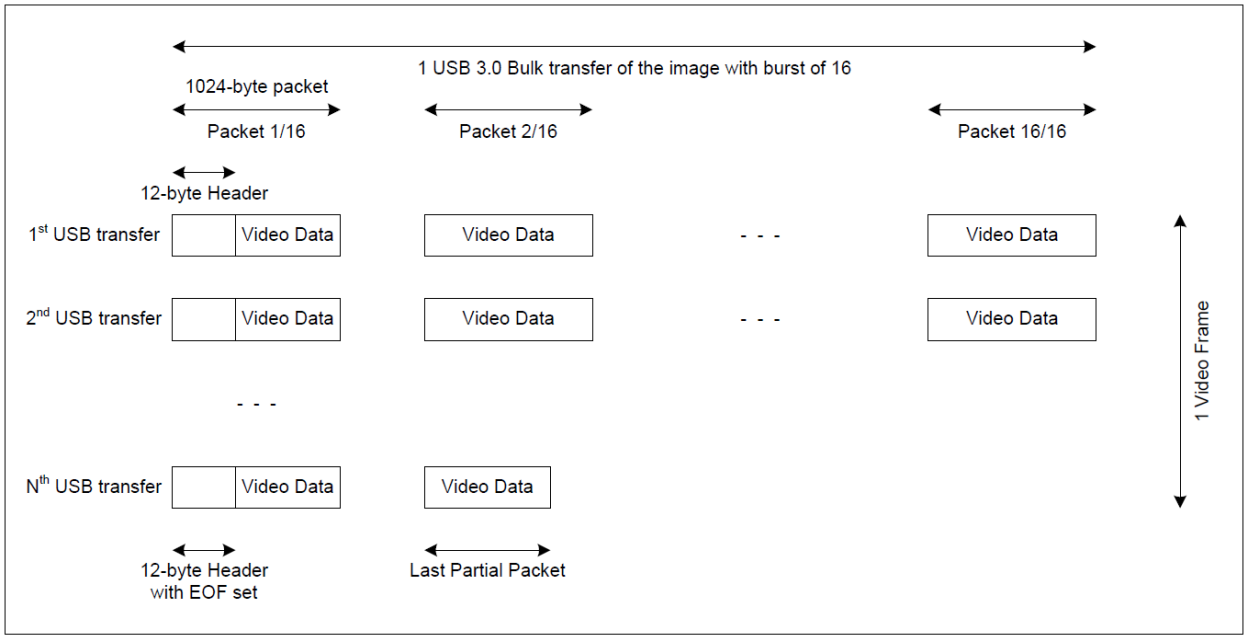


Figure 5.3.2-1 Illustration for the UVC Video Data Transfers

5.3.3 GPIF II State Machine

The GPIF – II peripheral is a part of the FX3 controller featuring a programmable Finite State machine – FSM able to be customized to drive the FX3 pins to interface with external hardware, such as an image sensor. To design the state machine, first is required to understand the interface requirements and FX3‘s DMA capabilities presented in more detail by the next topic. Here a more general approach will be used. The state machine design is illustrated next in figure 5.3.3-1.

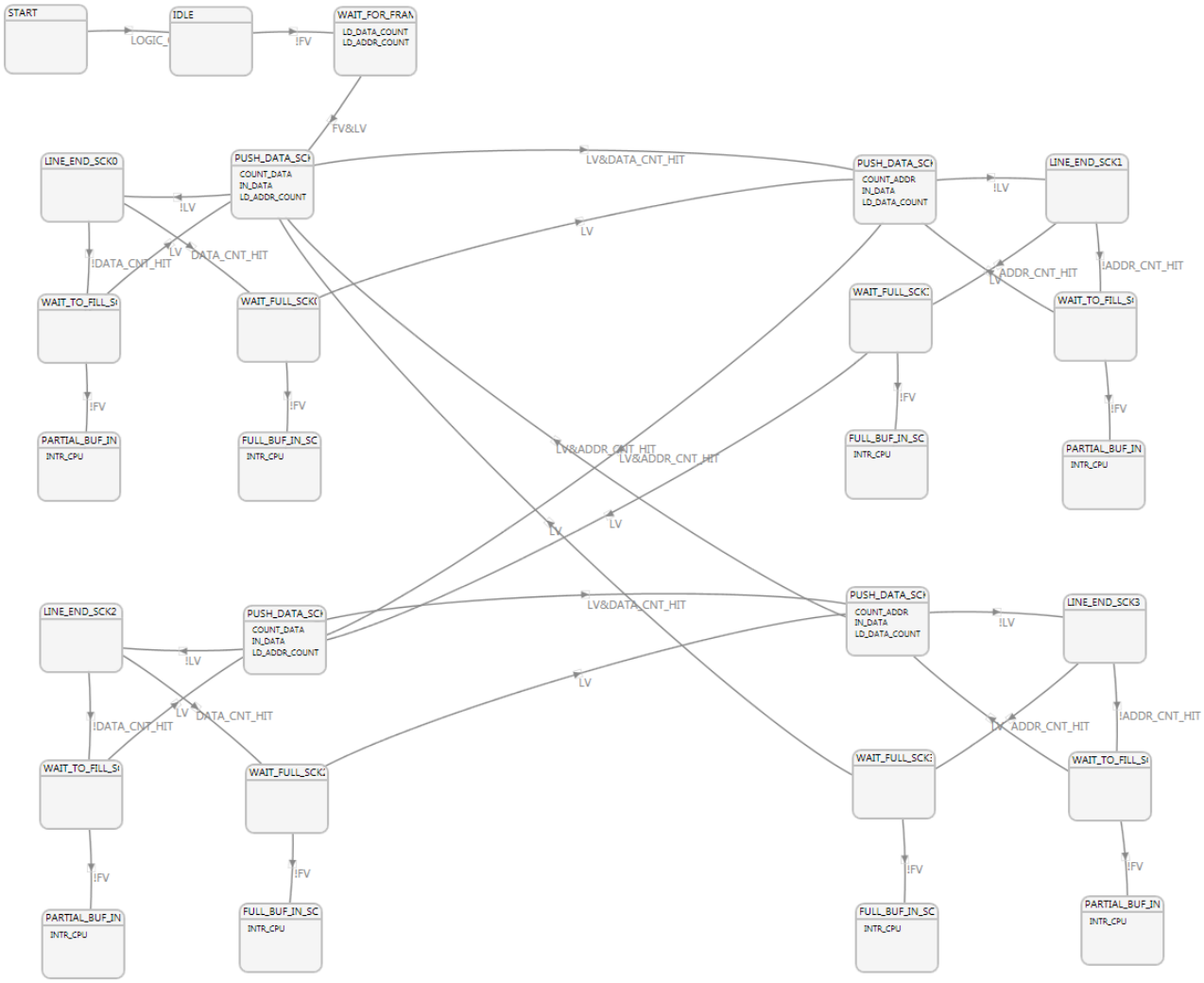


Figure 5.3.3-1 The Final State Machine Diagram Designed with GPIF – II Designer software.

To begin the GPIF II state machine, a point in the image sensor waveform (presented by Chapter 2.4) is been chosen for the state machine to start. The start of a frame, indicated by a positive FV transition, is a logical starting point. GPIF II detects this edge by first waiting for FV=0 (first state)

and then waiting for FV=1 (second state). The second state also initializes a transfer counter to correspond to one DMA buffer full of video data. The state machine tests the counter value and switches GPIF threads (DMA buffers) when the counter limit is reached. The counter limit is reached when a DMA buffer fills. The state machine uses two GPIF II internal counters to count DMA buffer bytes: the GPIF II address counter ADDR and the data counter DATA. Whenever the GPIF II state machine switches GPIF threads, it initializes the appropriate counter for the other GPIF thread. Because loading a counter limit takes one clock cycle, the loaded value is one less than the terminal count. The transfer counters increment by one every clock cycle. Therefore, depending on the data bus width of the interface, the value of the counter limit would change. For this example the data bus width is 8 bits and the DMA buffer size is 16,368 bytes. This is as a result of the USB protocol. The data burst in USB are done in a 16 x 1024 byte packets, or 16,384 bytes. The addition of a 12-byte header information regarding UVC and a padding of 4 bytes done by the ARM9 processor results to a total buffer size of 16,384-16=16,368 bytes. Minus the last the programmed limit should be 16,367. In general, the DMA buffer count limit is:

$$count = \left(\frac{producer\_buffer\_size(L)}{data\_bus\_width} \right) - 1$$

5.3.4 FX 3 DMA System

The GPIF II block is a part of the processor interface block – PIB and can run up to 100 MHz with 32 bits of data (400 MBps). To transfer the data into internal DMA buffers, GPIF II uses multiple GPIF threads connected to DMA producer sockets. All of the four GPIF threads are used by this design. An example of the threads and socket function is presented in figure 5.3.4.

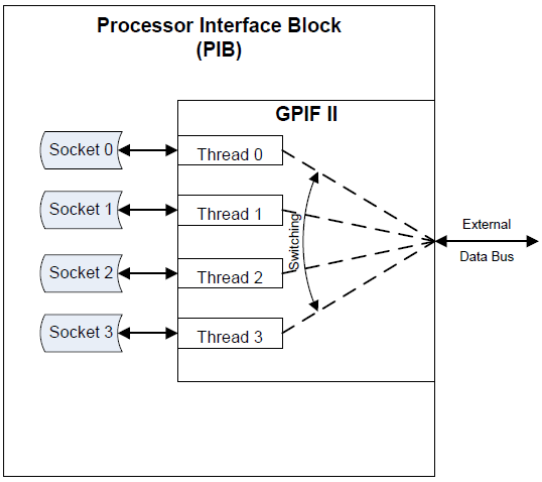


Figure 5.3.4-1 GPIF II Socket/Thread Mapping

To understand DMA transfers, the concept of a socket is further explored in the following four figures. Figure 5.3.4-2 shows the two main socket attributes, a linked list, and a data router.

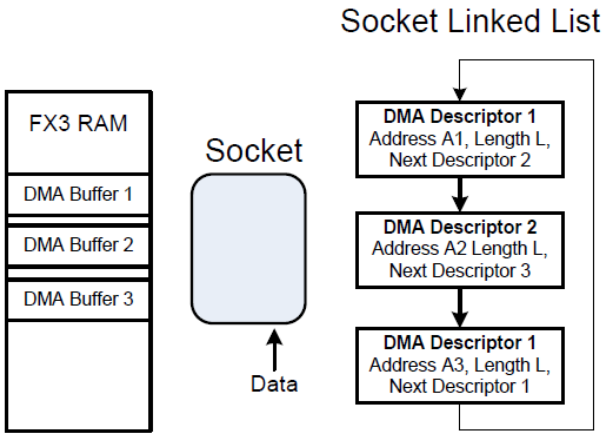


Figure 5.3.4-2 A Socket Routes Data According to a List of DMA Descriptors.

The Socket linked list is a set of data structures in main memory called DMA Descriptors. Each Descriptor specifies a DMA buffer address and length as well as a pointer to the next DMA Descriptor. As the socket operates, it retrieves the DMA Descriptors one at a time, routing the data to the DMA buffer specified by the Descriptor address and length. When L bytes have been transferred, the socket retrieves the next Descriptor and continues transferring bytes to a different DMA buffer. This structure makes a socket extremely versatile because any number of DMA buffers can be created anywhere in memory and be automatically chained together. For example, the socket in Figure 5.3.4-2 retrieves DMA Descriptors in a repeating loop.

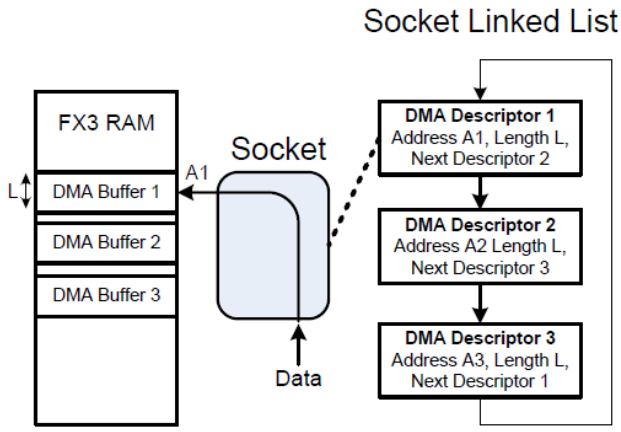


Figure 5.3.4-3 A Socket Operating with DMA Descriptor 1

In Figure 5.3.4-3, the socket has loaded DMA Descriptor 1, which tells it to transfer bytes starting at A1 until it has transferred L bytes, at which time it retrieves DMA Descriptor 2 and continues with its address and length settings A2 and L (Figure 5.3.4-4 below).

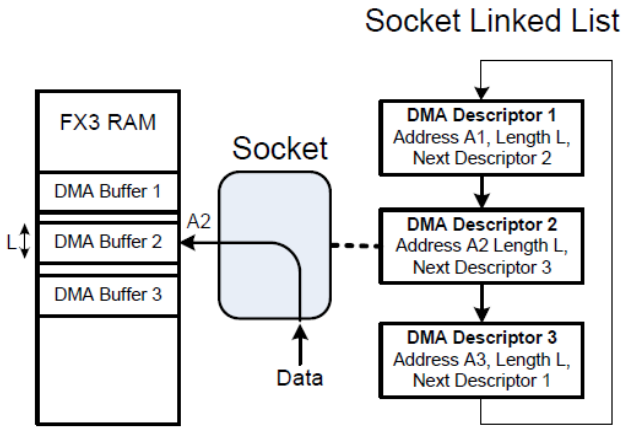


Figure 5.3.4-4 A Socket Operating with DMA Descriptor 2.

In Figure 5.3.4-5 below the Socket retrieves the third DMA Descriptor and transfers data starting at A3. When it has transferred L bytes, the sequence repeats with DMA Descriptor 1.

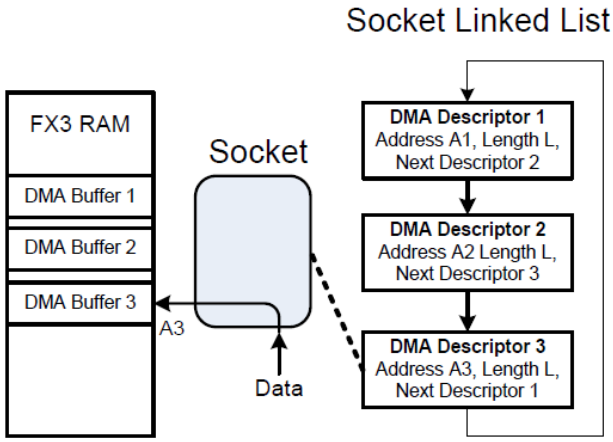


Figure 5.3.4-5 A Socket Operating with DMA Descriptor 3.

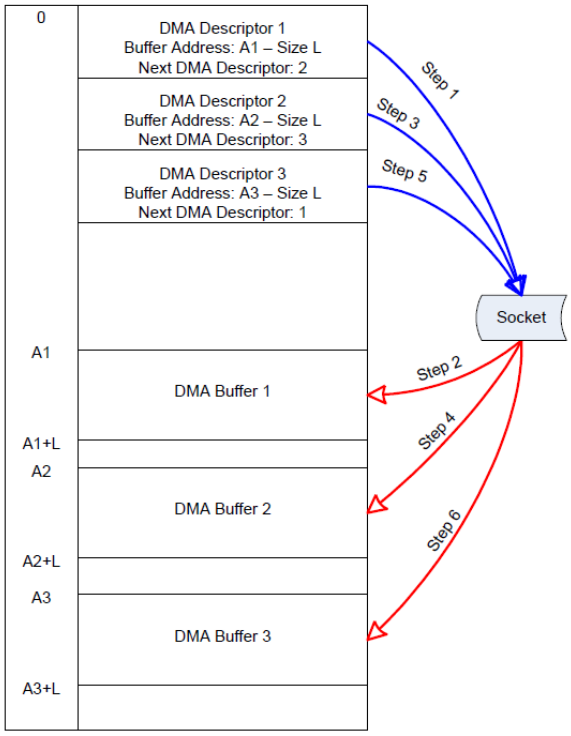


Figure 5.3.4-6 DMA Transfer Example.

Figure5.3.4-6 above shows a DMA data transfer in more detail. This example uses three DMA buffers of length L chained in a circular loop. FX3 memory addresses are on the left. The blue arrows show the socket loading the socket linked list Descriptors from memory. The red arrows show the resulting data paths. The following steps show the socket sequence as data is moved to the internal DMA buffers:

- Step 1: Load DMA Descriptor 1 from the memory into the socket. Get the DMA buffer location (A1), DMA buffer size (L), and next Descriptor (DMA Descriptor 2) information. Go to step 2.

- Step 2: Transfer data to the DMA buffer location starting at A1. After transferring DMA buffer size L amount of data, go to step 3.

- Step 3: Load DMA Descriptor 2 as pointed to by the current DMA Descriptor 1. Get the DMA buffer location (A2), DMA buffer size (L), and next Descriptor (DMA Descriptor 3) information. Go to step 4.

- Step 4: Transfer data to the DMA buffer location starting at A2. After transferring DMA buffer size L amount of data, go to step 5.

- Step 5: Load DMA Descriptor 3 as pointed to by the current DMA Descriptor 2. Get the DMA buffer location (A3), DMA buffer size (L), and next Descriptor (DMA Descriptor 1) information. Go to step 6.

- Step 6: Transfer data to the DMA buffer location starting at A3. After transferring DMA buffer size L amount of data, go to step 1.

This simple scheme has an issue in the camera application. A socket takes time to retrieve the next DMA Descriptor from memory, typically 1 microsecond. If this transfer pause occurs in the middle of a video line, the video data is lost. To prevent this loss, the DMA buffer size can be set as a multiple of the video line length. This would make the DMA buffer switching pause coincide with the time that the video line is inactive (LV=0). However, this approach lacks flexibility if, for example, the video resolution is changed.

Setting DMA buffer size exactly equal to line size is also not a good solution because it does not take advantage of the USB 3.0 maximum burst rate for BULK transfers. USB 3.0 allows a maximum of 16 bursts of 1024 bytes over BULK Endpoints. This is why the DMA buffer size is set to 16 KB.

A better solution is to take advantage of the fact that sockets can be switched without latency in one clock cycle. Therefore, it makes sense to use four Sockets to store data into four interleaved DMA buffers. The use of tow sockets can be exploited when the firmware runs a more simple video transfer protocol like YUY2, requiring less efficiency in data transfers. For the RGB24 transfer protocol the application note firmware does not perform well and additional sockets are needed. Data transfer using dual sockets is described in Figure 5.3.4-8, again with numbered execution steps. Socket0 and Socket1 access to DMA buffers is differentiated by red and green arrows (data paths for individual sockets), respectively. The a and b parts of each step occur simultaneously. This parallel operation of the hardware eliminates DMA Descriptor retrieval dead time and allows the GPIF II to stream data continuously into internal memory. These steps correspond to the “Step” line in Figure 5.3.4-7 below.

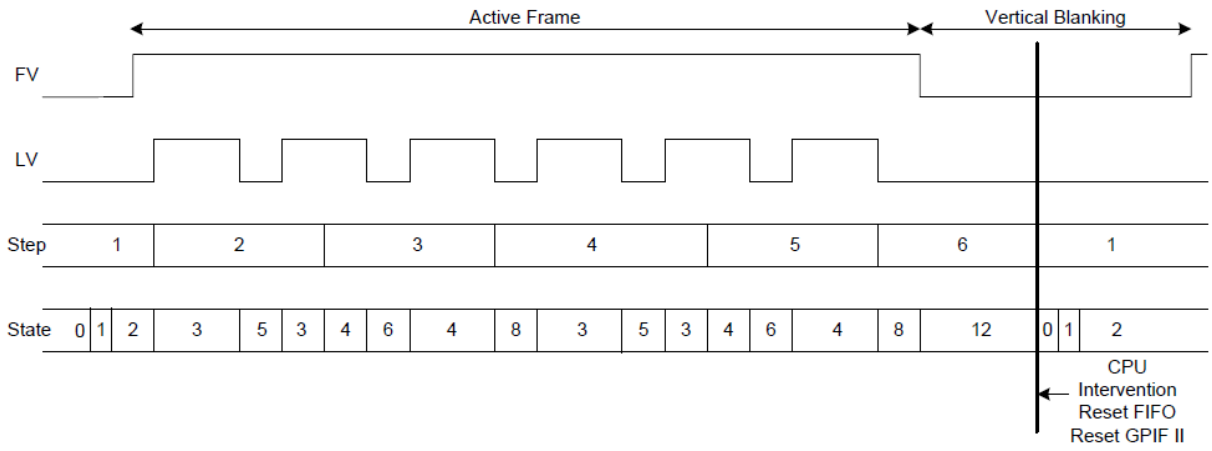


Figure 5.3.4-7 Image Sensor Interface, Data Path Execution, and State Machine Correlation.

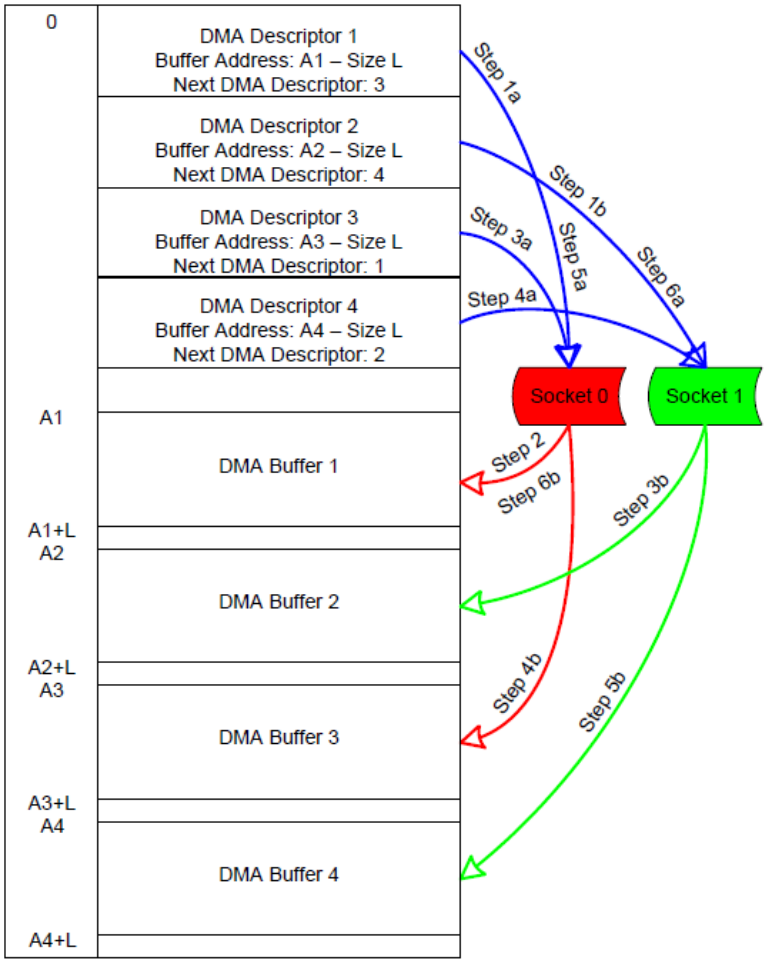


Figure 5.3.4-8 Dual Sockets Yield Seamless Transfers.

- Step 1: At initialization of the sockets, Socket 0 and Socket 1 load the DMA Descriptor 1 and DMA Descriptor 2, respectively.
- Step 2: As soon as the data is available, Socket 0 transfers the data to DMA buffer 1. The transfer length is L. At the end of this transfer, go to step 3.
- Step 3: GPIF II switches the GPIF thread and, therefore, the socket for data transfer. Socket 1 starts to transfer data to DMA buffer 2, and, at the same time, Socket 0 loads the DMA Descriptor 3. By the time Socket 1 finishes transferring L amount of data, Socket 0 is ready to transfer data into DMA buffer 3.
- Step 4: GPIF II now switches back to the original GPIF thread. Socket 0 now transfers the data of length L into DMA buffer 3. At the same time, Socket 1 loads the DMA Descriptor 4, making it ready to transfer data to DMA buffer 4. After Socket 0 finishes transferring the data of length L, go to step 5.
- Step 5: GPIF II routes Socket 1 data into DMA buffer 4. At the same time, Socket 0 loads DMA Descriptor 1 to prepare to transfer data into DMA buffer 1. Notice that Step 5a is the same as Step 1a except that Socket 1 is not initializing but, rather, transferring data simultaneously.
- Step 6: GPIF II switches sockets again, and Socket 0 starts to transfer data of length L into DMA buffer 1. It is assumed that by now, the DMA buffer is empty, having been depleted by the UIB consumer socket. At the samet ime, Socket 1 loads the DMA Descriptor 2 and is ready to transfer data into DMA buffer 2. The cycle now goes to Step 3 in the execution path.

GPIF II Sockets can transfer video data only if the consuming side (USB) empties and releases the DMA buffers in time to receive the next chunk of video data from GPIF II. If the consumer is not fast enough, the sockets drop data because their DMA buffer writes are ignored. As a result, the byte counters lose sync with the actual transfers, which can propagate to the next frame. Therefore, a cleanup mechanism is required at the end of every frame.

At the end of a frame, the GPIF II state machine generates a CPU interrupt, which starts the chain of events described above. The firmware waits for the USB side to drain the DMA buffer data using a loop in the UVCAppThread_Entry function. As soon as hitFV (which is set by the GPIF callback function) is set, the last DMA buffer is committed to USB, and the prodCount equals the consCount, the firmware will do the following:

- Reset the DMA channel (which resets its FIFO).
- Toggle the UVC header FID bit.
- Call the CyU3PGpifSMSwitch function in order to restart the GPIF II state machine at its START state.
- Wait for FV to go HIGH again.

6 PERFORMANCE EVALUATION

This chapter will examine the final results in image quality delivered by the imaging system.

6.1 Demosaicing

As examined by Chapter 2.5 the processing of pixel RAW intensity data done by the demosaicing algorithms can result in different image quality. The algorithms involved are the bilinear interpolation demosaicing and the Contour Stencils demosaicing algorithms. The result is an estimation of the missing values of the two bands that have been cut off by the Bayer color filter array – CFA.

The recovered pixel intensity data are then used by the algorithm to estimate the missing values in different ways. The result is a full colored image of the object with all the information of the present pixel bands and the estimated values.

Bilinear interpolation demosaicing: This algorithm is a Matlab implementation that is fully optimized for Matlab usage. The algorithm runs at the arrival of each frame from the UVC and project the produced image. The speed of the code enables frame rates up to 60 fps. Unfortunately the firmware code needs more optimizations to achieve frame rates higher than 60 fps for 720p resolution and the YUY2 video transfer protocol. The GPU of the host pc can run the code much faster but the GPU must be compatible for the parallel toolbox.

Contour Stencils demosaicing: This demosaicing algorithm runs offline because of the extensive computational power required. A more specialized hast application written in a modern language like C# and C++ can result in a realistic execution for the Contour Stencils demosaicing algorithm. Furthermore GPU optimizations can increase application performance and maintain image quality at the same time.

The Contour Stencils demosaicing algorithm calculates the missing intensity values of each band with respect to the edges of the image. That means the result image is a sharper interpretation of the actual object than the image produced by the bilinear demosaicing algorithm. The code[15] is compiled and runs offline. The result is an image like the one shown in figure 6.1-1.

[15] http://www.ipol.im/pub/art/2012/g-dwcs/

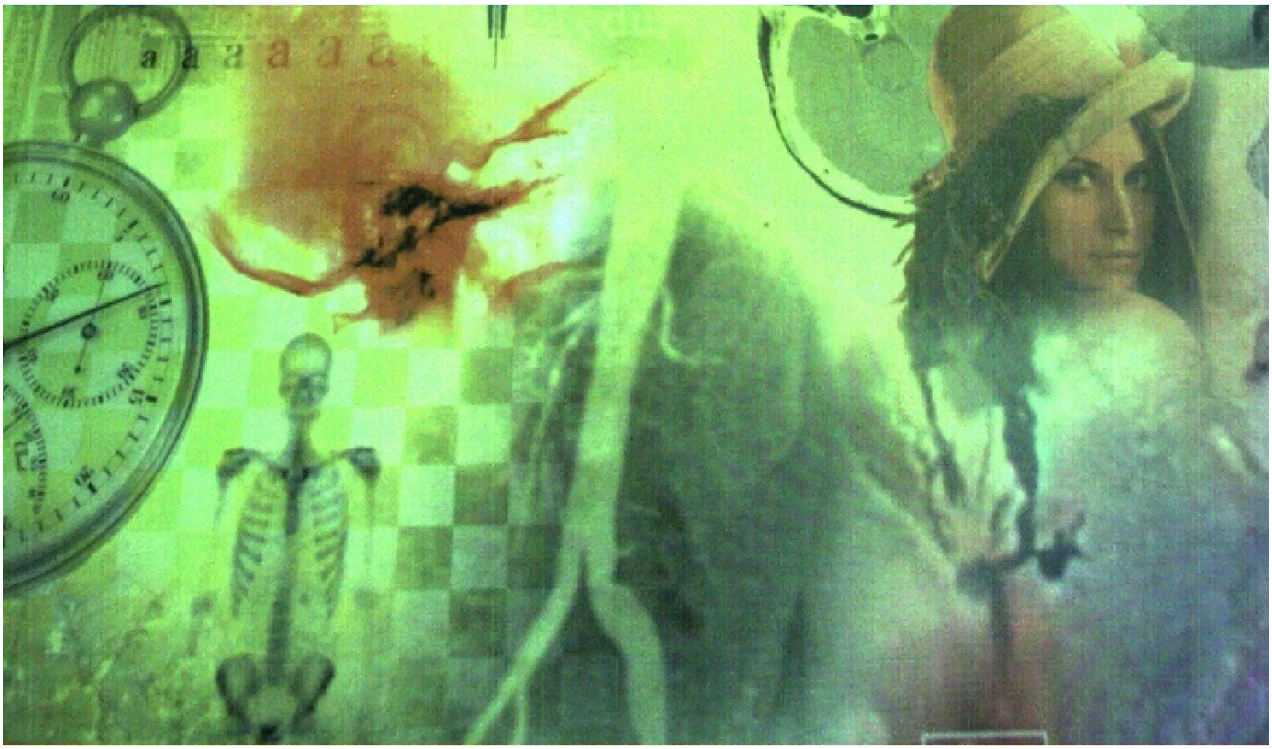


Figure 5.3.4-1 Contour Stencils demosaicing done offline. The image is taken from the book hardcover: Digital Image Processing (3rd Edition) [6.1].

The above image is a 720p (1200 x 720) resolution from a video frame captured by the developed imager. The initial pixel values of a single frame was introduced to the algorithm and then the CFA pattern ('grrb') was also given as input. The bilinear interpolation done by Matlab in real time were also impressive with a severer performance advantage over Contour Stencils. To be fair the bilinear interpolation code that Matlab runs is an optimized version created for execution by this kind of applications and poses no performance issue. On the other hand for the Contour Stencils algorithm implementation a gcc compiler is needed and cannot be introduced to Matlab in an efficient way.

The comparison between the two demosaicing algorithms can be done even visually but a more scientific method is required.

Figure 6.1-2 presents the final images produced by the bilinear interpolation and Contour Stencils algorithms as a detail from the image presented earlier. Along with them, the step images of RAW pixel data and the mosaic pattern created by the application in Matlab are also included.

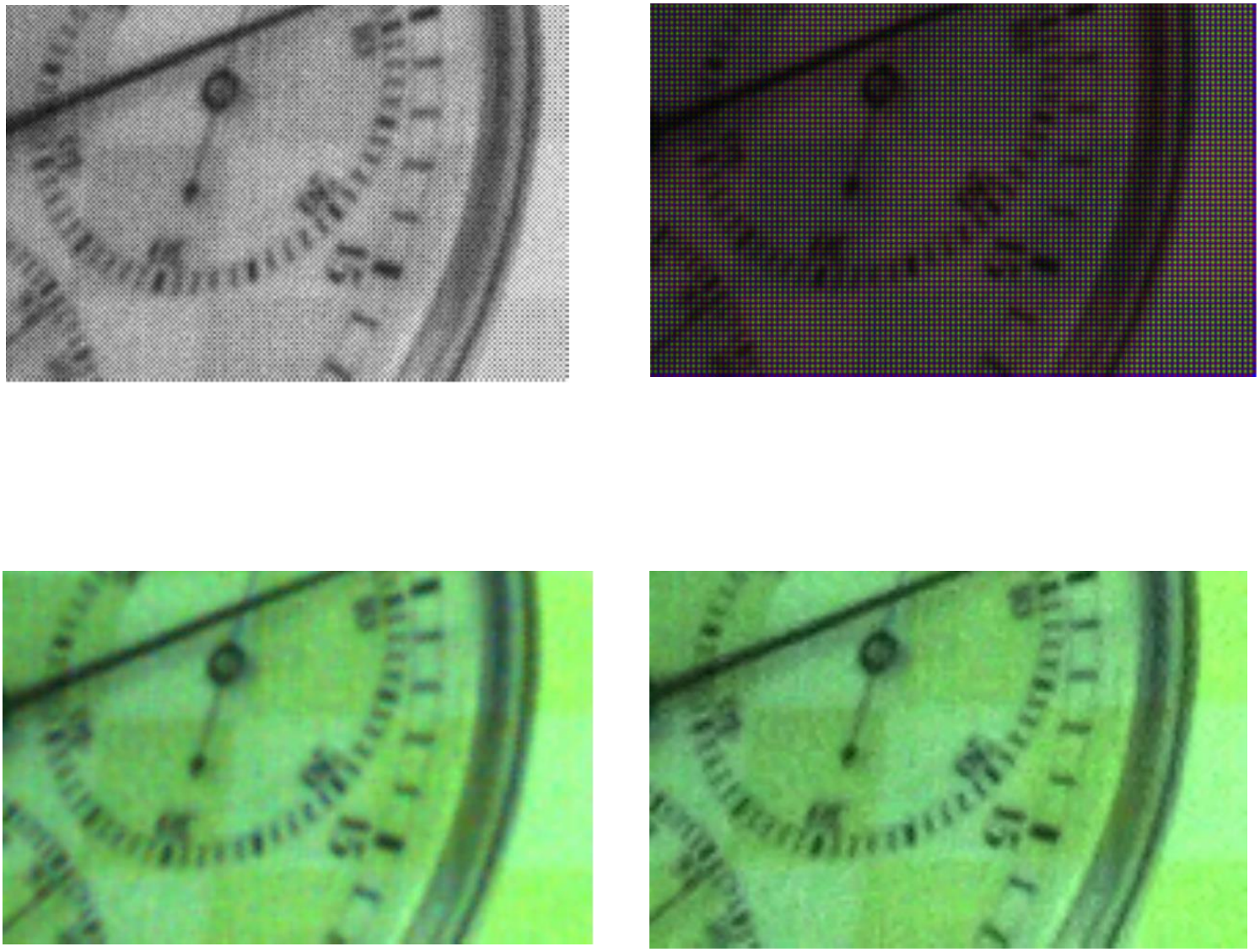


Figure 5.3.4-2 A detail taken from the first image indicating the differences between the various stages of the frame processing and a closer look to the performance of the bilinear interpolation demosaicing. First at the up left corner is the RAW pixel values taken from the imager. The second image on the up right corner presents the "colored" RAW pixel intensity values according to the Bayern CFA. At the bottom the bilinear interpolation demosaicing (left) and the Contour Stencils algorithm (right) are presented.

6.2 System Spatial Resolution

Here the procedure and final results for the scientific image Resolution are presented. The difference between a casual image and a scientific image is the fact that the first provides a final result that satisfies the human eye [6.2] where the second method is more consistent regarding the scientific data that can be extracted from the image.

6.2.1 Modulation Transfer Function – MTF Measurement

As described in Chapter 2.7.1, the MTF shows the ability of the imaging system to capture the details of an object. In other words, it measures how accurate the system reproduces (or transfers) detail from the object to the final image.
To measure MTF a Grill/Square-Wave Analysis Pattern is introduced as target for the imaging system. The images taken are introduced as input to the software discussed in Chapter 2.7.2 and the following situations resulting in different image resolution will be evaluated:

- Image after demosaicing with bilinear interpolation algorithm.
- Image after process by the Contour Stencils demosaicing algorithm with alpha parameter equal to 1.2 (see Chapter 2.5)
- Image with Contour Stencils alpha parameter set to 2.
- The RAW pixel intensities without any demosaicing process.

The resulting grill pattern images of the previously mentioned methods are presented in figures 6.2.1-(1, 2, 3, and 4) to the next page.

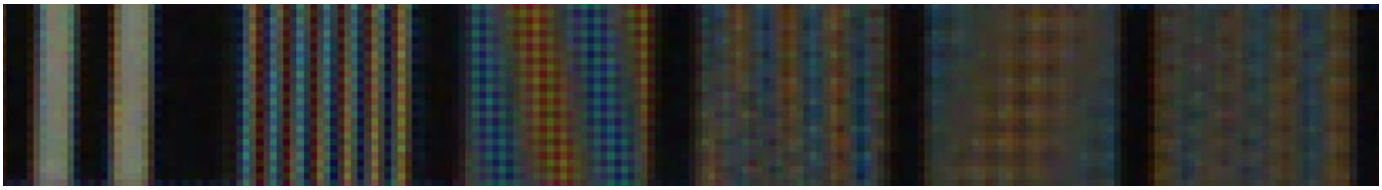


Figure 6.2.1-1 Pattern after Bilinear interpolation demosaicing

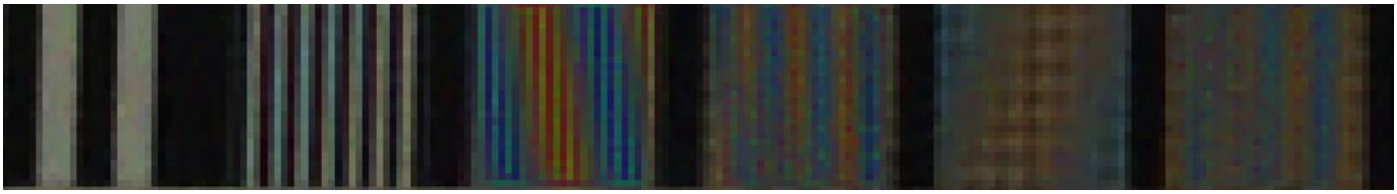


Figure 6.2.1-2 Pattern after Contour Stencils demosaicing algorithm with alpha parameter equal to 1.2

Figure 6.2.1-3 Pattern after Contour Stencils with alpha parameter set to 2.

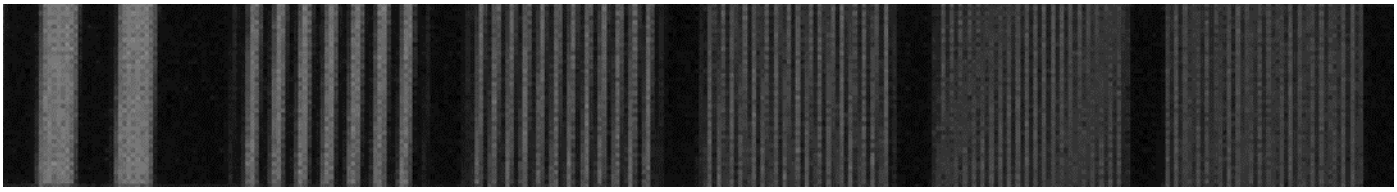


Figure 6.2.1-4 Pattern produced by the RAW pixel intensities without any demosaicing process.

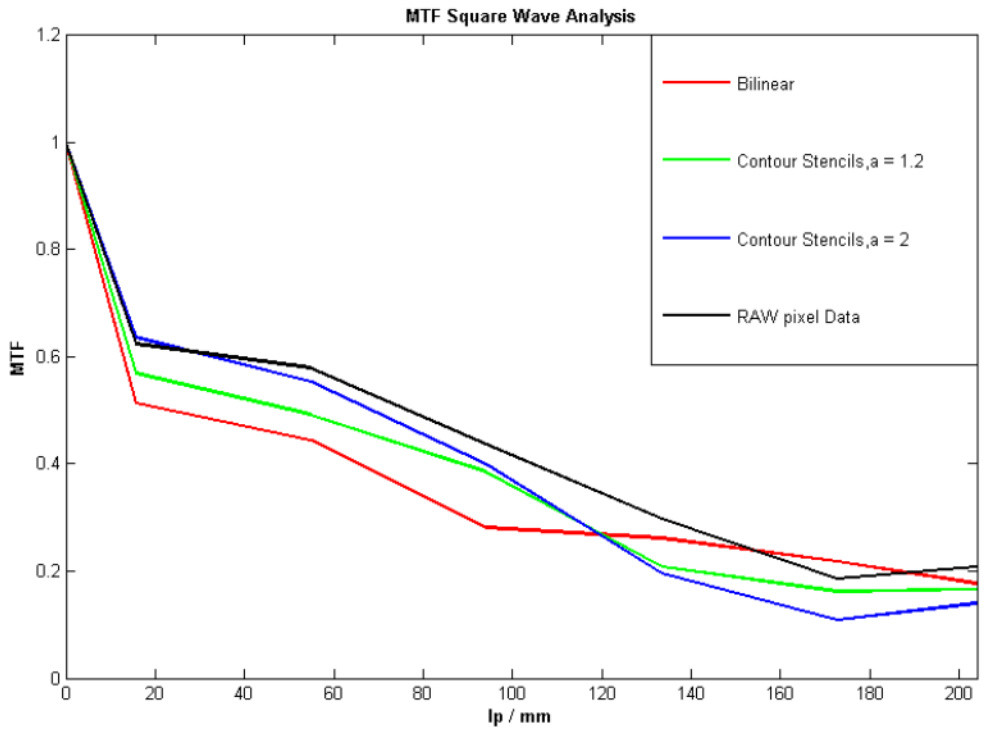
Next the required information about the optical system recovered from chapter 5.2.3 are set as parameters to the software. More precisely the lens aperture. Then from the sensors data sheet the pixel size and pitch (dead space between pixels, see Chapter 2.3.2) are taken into consideration.
Next the pixels covering each line pair are clearly visible in a magnification of the RAW pixel data pattern. The same information is going to be used thru the measurement of all four images.
Important characteristic from the calculations in the Nyquist frequency. The Nyquist frequency is the frequency where (measured in lp/mm) a sensor pixel has a line pair in it, therefore the line pair is not visible and the Nyquist frequency is called cut – off frequency.

The sensor features a 3.75 µm pixel size therefore the Nyquist frequency measured is at 133.3 lp/mm. More precisely the cut-off frequency is:

$$f_N = 0.5 \ {}^{lp}/_{pixel}$$

For a pixel at 3.75µm: $f_N = 133.3 \ {}^{lp}/_{mm}$

The MTF measurement results are shown in figure 6.2.1-5 below.



These results are expected as they agree with the visual images. Near f_N at 133.3 ${}^{lp}/_{mm}$ the measurements are getting confusing as a result of a crucial measurement error.
The zipper artifacts caused by slightly tilt of the target case aliasing before the Nyquist frequency is reached.
This error is unavoidable because of the fine detail and accuracy of the error. The pixels are introduced to light coming from an inaccurate target causing this phenomenon.

7 DEVELOPED APPLICATIONS

In this chapter some of the applications developed using the imaging system will be discussed.

7.1 Imaging at low light conditions

The MT9M034 1.2MP Image sensor performs very well under low light conditions but as a FSI sensor (see Chapter 2.3) introduce some noise artifacts that are more intense in comparison with a BSI sensor.

Although the imaging at low light conditions require a long exposure time for the photosensitive pixel array of the image to collect the necessary photons for a good image.

Increasing the integration time introduce some strong noise mainly caused by temperature. Temperature cases dark current and shot noise artifacts which can be seen in an image like “hot” pixels. Some of this noise is also caused by Fix Pattern Noise – FPN introduced by the uniformity of the pixel array. Each pixel adds different amount of noise variation casing a repeating pattern. When pixels are required to perform extreme amplification of the collected charge, the FPN increases dramatically.

If this kind of noise is present in each frame the same way, it can be captured and then subtracted from the frame resulting in an almost noise free frame.

The noise is captured in a “Dark Frame” taken with the optics covered and exactly right before and after the actual image frame has been captured. The tow Dark frames are averaged and the new dark frame contains an accurate estimation of the FPN in the frame been captured.

The main idea behind this approach is that the noise present in the frames has not been changed in any way with ISP or demosaicing resulting in loss of information. We can say that the noise in the Dark frame captured and the noise in the actual frame are more associated in a 2-d image than after the demosaicing has predicted the missing pixel values.

The final image is the result of an extended exposure time. This is achievable by setting the frame rate at extremely low rates. A register of the CMOS sensor can change the frame rate by assigning a value for the horizontal and vertical blanking shown by chapter 5.2.1. Then the only parameter needed for a long exposure is the integration time. The integration time is controlled by the same parameter controlling the “Brightness” variable visible in the host application. Setting this parameter to 0 gives an invalid assignment to the register causing it to change its value to the maximum possible. Alternatively any other value can be set thru Matlab by disabling the range check in the firmware.

Below the result image captured with a total integration time of 40 seconds shown by figure 7.1.1-1.

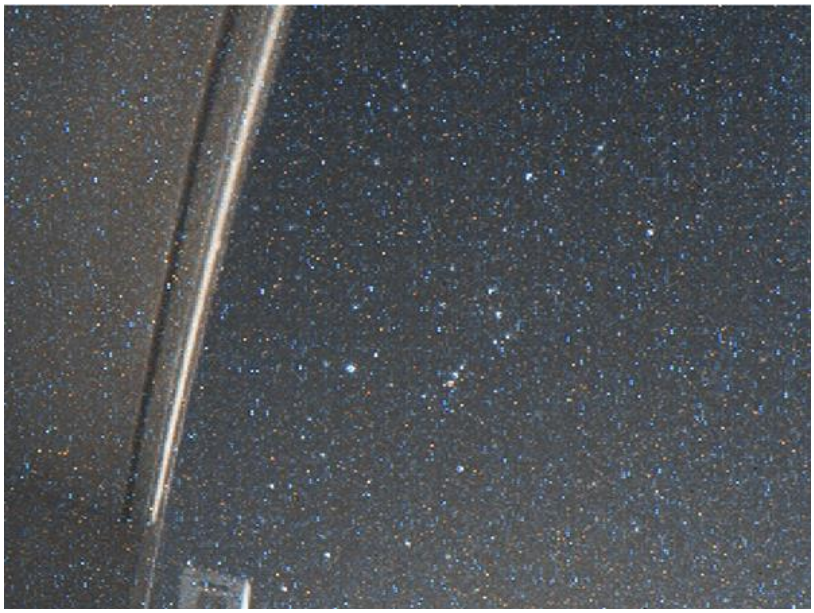


Figure 6.2.1-1 : The result of a long exposure image captured by the CMOS sensor.

A “dark frame” was already been captured and a second one is captured after the image above. Then the tow of them are added and then divided by 2 in order to average the difference resulting in micro changes of the noise while the actual image was integrating. The result of the 2 averaged frames is shown by figure 7.1.1-2.

Figure 6.2.1-2 The result of 2 averaged “dark frames”

Both images are shown after the demosaicing but the procedure is done with the RAW pixel data and not with the images presented above.
The strong noise present in each image is relatively the same intensity and follows the same patterns. This way if we subtract the one from the other the result is a more noise free image. The final result after this procedure is shown by figure 7.1.1-3.

Figure 6.2.1-3 Final result of the de-noise procedure with dark frame.

The results shown by the image above are very impressive considering the small lens diameter and moderate optics quality. In addition the Orion nebula (see figure 7.1-4) in the captured image is shown as a white spot. This is the result of the low pass infrared filter embedded into the optical system of the CMOS camera board. If a proper optical system is introduced, most of the deep sky object emissions in the Near Infra-Red – NIR spectrum will be captured as red light resulting in a better color in image. With this detail in mind, the condor Stencils demosaicing with high alpha value is used for better image quality.

7.2 Real Time Human Vision Emulation

The human visual system can be regarded as consisting of two parts. The eyes act as image receptors which capture light and convert it into signals which are then transmitted to image processing centers in the brain. The human eye senses light through a thin tissue placed at its back end; the retina. The retina acts pretty much like a film and it consists of two main categories of cells: nerve cells and light-sensitive cells. There are two general classes of light sensitive cells; rods and cones. Rod cells are very sensitive and provide visual capability at very low light levels. Cone cells perform best at normal light levels.

An interesting fact about the retina is that it is not uniformly light-sensitive; this is attributed to the fact that the distribution of rods and cones is not uniform across the retina. In figure 7.2.1-1 the photoreceptors of the human eye are presented.

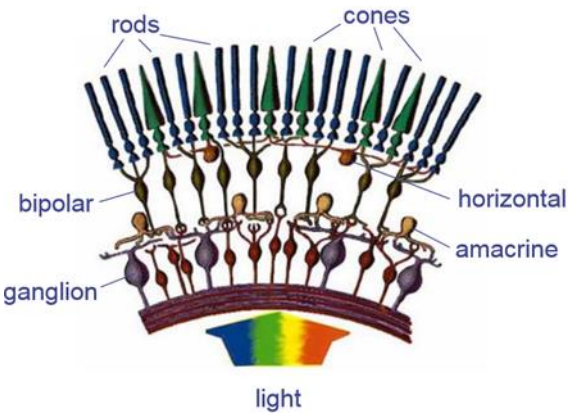


Figure 6.2.1-1 The "pixels" of the human eye.

Observing figure 7.2-1 we can see that the robs are connected to the same "wire" – the bipolar. Cones on the other hand are connected individually resulting in a more detailed image. This function is very similar to binning performed by the CMOS sensor. Moreover the "pixels" of the human eye have one more characteristic, they are not uniformly distributed as illustrated by figure 7.2.1-2.

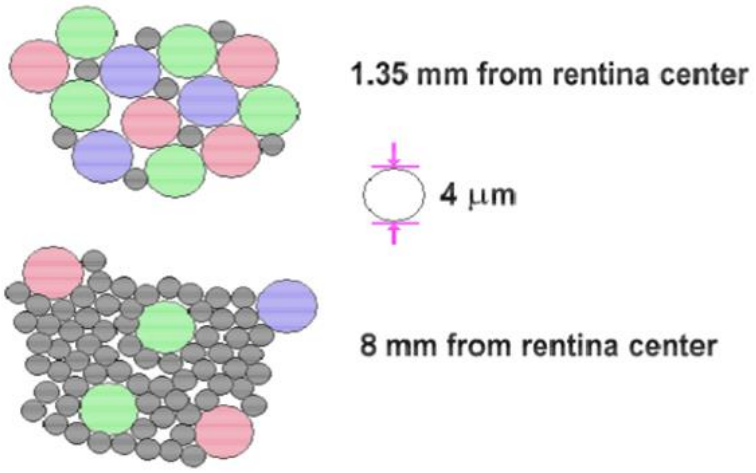


Figure 6.2.1-2 The spatial distribution of cones and robs.

With that in mind a biologically inspired application has been developed – The human vision emulation. To emulate the HVS, the sensor must take advantage of the binning in order to achieve better light sensitivity and reduce image data for processing. This must be applicable for the pixel array around a certain area. This area functions very similar to the retina center offering better resolution.

The approach behind this application is that the sensor can read a binned frame of all 1.2MP (1200X1980) and give it as a VGA frame. This frame is the full pixel array reading from the sensor. Then after the overview frame is captured the sensor changes context to meet the requirements of a new frame captured without binning at the center of the pixel array. This frame offers data from each pixel individually, pretty much like the cone in the human eye.

Then the tow frames are transferred by the FX3 at the high frame rate of 60 fps. Then the frames are captured by the host application and the actual frame is been created.

In order to distinguish which frame is which, the application calculates the correlation and the structural similarity index of frames in the buffer and compares them to determine if the received frame is high resolution or an overview frame.

$$corr2(A,B) = \frac{\sum_M \sum_N (A_{MN} - \overline{A})(B_{MN} - \overline{B})}{\sqrt{\sum_M \sum_N \left(A_{MN} - \overline{A}\right)^2 * \sum_M \sum_N \left(B_{MN} - \overline{B}\right)^2}}$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1) * (2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) * (\sigma_x^2 + \sigma_y^2 + c_2)}$$

The result of this algorithm is a frame with a low resolution in the surrounding and a higher resolution at the center (example below).

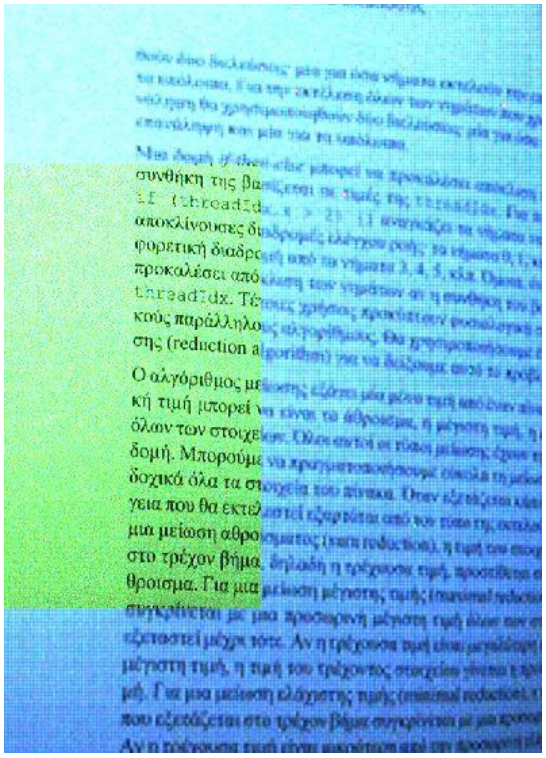


Figure 6.2.1-3 Detail of the received frame from the human vision application.

As observed in figure 7.2.1-3 the text in the surrounding area is much more blurred, the result of binning. On the other hand the text in the center area is a 1080p resolution frame given at a VGA connection!

One important aspect also observed is the intensity accuracy of the surrounding area because of the higher light performance of the binned pixels (much like the robs).

One drawback of this application is the frame rate. Although the FX3 controller and the imager give 60fps, matlab performs poorly resulting in frame loss and stalling the frames received in order to correct this error.

One more problem faced is the miscalculation of high resolution frames as overview frames. This mistake is done because the corr2 and SSIM algorithms failed. This failure can be caused by two reasons. First a uniform area been misinterpreted as false similarity with the same kind of frame. The other reason is that matlab simply lost all of the frames of one kind due to low performance and is comparing two frames of the same kind. Fortunately this problem is presented only for one frame and is corrected right after the next frame is been received.

One artifact presented in a higher frame rate is the fixed pattern noise - FPN presented when the sensor reads at a high frame rate. This problem is fixable thru column correction done by the image sensor itself. When context switch operations take place the column correction is required. One example of FPN at a very high frame rate of 120fps is presented by figure 7.2.1-4. This fact limits the frame rate of the sensor for context switch operations and a more specialized driver for the camera sensor is needed.

Figure 6.2.1-4 FPN presented when the imager operates at high frame rates.

8 CONCLUSIONS

As a result of the market research for consumer available products, a platform design built around the FX3 controller was discovered. The design presented by this thesis is just a small portion of the capabilities and connectivity presented by this hardware. In the next few paragraphs, a detailed summary of the observations and conclusions regarding the various sensor implementation, the FX3 performance and firmware development and future development of a complete platform design will be discussed. Also an overview of the system performance presented will be discussed.

8.1 Sensor Connectivity

The implementation of an image sensor for this design was relatively easy, with some minimal technical problems regarding the physical design of the adaptor mainly caused by the prissy design of the molex connector. The use of other imagers is possible by changing the adaptor created for the LI-CAM-M034 camera board. One issue is the lack of front end electronics for the most imagers available. The design of a new PCB featuring some electronics for the camera front end is mandatory if a CMOS chip is available for implementation.

One detail that needs to be mentioned is the lack of support from the image sensors manufacturers. Mainstream image sensors are hard to acquire as a single chip and the manufacturers communicate only via B2B (business to business) context. Moreover the design of electronics for the front end is a complicated procedure and a more dedicated study is needed.

For the moment, third party manufacturers like Leopard imaging can produce camera boards featuring the front end electronics.

8.2 FX3 Controller and firmware development

FX3 peripheral controller is the result of many months of research and discarded solutions (see Chapter 4). The results from the FX3 peripheral controller performance are impressive. The firmware implementation was relatively easy and the support from Cypress flawless, providing all the technical information needed and featuring online forums with professional help. These facts makes the FX3 controller a perfect solution for this design and has many unexploited features for other applications, like the support of a custom video class. Access to RAW pixel data from the imager increased frame rates and offered new ways of image processing.

The main advantage of the controller is its ability to interface with the image sensor or FPGA giving the ability to integrate a bridge between CMOS sensor and the FX3.

8.3 Overview of the system performance

This thesis have achieved the targeted objective, and gave an example of the capabilities and advantages of a smart imager design. The decision for off board processing and ISP algorithms offered new opportunities for new applications. This approach can offer much more in new applications.

The weak link of the developed imaging system was the host application. Matlab is a computationally heavy program and it compiles code line by line like an interpreter, slowing framerates and performance.

The access to the RAW image data offered new opportunities for image processing and increased image quality.

9 FUTURE WORK

Any future work associated with this thesis is divided into two sections: the future work regarding this design and the future work that can be done to develop a new imaging platform.

9.1 This design

The development of a new firmware for giving more control over the image sensor it's a safe way to start. Then a new image sensor can be implemented for exploiting new technologies like the BSI.

Furthermore an FPGA (from Altera or Xilinx etc.) can interface with the CYUSB3KIT-003 EZ-USB® FX3™ providing connectivity and interface capabilities for the serial connection protocol imagers featuring subLVDS or MIPI CSI-2.

A new application can also been developed written in a modern programming language like C++ or C# for a significant performance boost and new features.

9.2 A Platform Design

New hardware featuring the FX3 is available for integration into a larger and more flexible design. The hardware proposed for such design is the Lattice USB 3 Board and if possible to acquire the Net Vision SVI board.

10 References

[2.1] Costas Balas : HPY 603 (2014) Electronic imaging: Advanced topics and applications

[2.2] Costas Balas : HPY412 (2013) Optoelectronics, Lecture 2

[2.3] Costas Balas : HPY 204 (2012) Electronics I , Lecture 6

[2.4] *Alireza Moini : Centre for High Performance Integrated Technologies and Systems (CHIPTEC)*

[2.5] Axis Communications: CCD and CMOS sensor technology Technical white paper

[2.6] Mortimer Abramowitz, Michael W. Davidson: Anatomy of a Charge-Coupled Device

[2.7] Learning center material by andor.com: CCD Sensor Architectures

[2.8] Edward R. Dougherty: Electronic Imaging Technology

[2.9] Thomas J. Fellers and Michael W. Davidson: CCD Saturation and Blooming

[2.10] Dr. Judi Provencal, 268 Sharp Lab: Detectors in Astronomy

[2.11] Junichi Nakamura: Image Sensors and Signal Processing for Digital Still Cameras

[2.12] Gözen KÖKLÜ: Design and Implementation of CMOS Image Sensors for Biomedical Applications (June 2014)

[2.13] Xinqiao Liu: CMOS IMAGE SENSORS DYNAMIC RANGE AND SNR ENHANCEMENT VIA STATISTICAL SIGNAL PROCESSING (June 2002)

[2.14] Eric R. Fossum, Fellow, IEEE, and Donald B. Hondongwa, Student Member, IEEE: A Review of the Pinned Photodiode for CCD and CMOS Image Sensors

[2.15] E. R. Fossum, "Active Pixel Sensors: Are CCDs dinosaurs?" in Proc. SPIE CCD's Optical Sensors III , vol. 1900. 1993, pp. 2–14.

[2.16] E. R. Fossum, S. K. Mendis, and S. E. Kemeny, "CMOS active pixel image sensor," U.S. Patent 5,471,515, Nov. 28, 1995.

[2.17] E. R. Fossum, "Ultra low power imaging systems using CMOS image sensor technology," in Proc. SPIE Adv. Microdevices Space Sci.Sensors, vol. 2267. 1994 pp. 107–111.

[2.18] T. H. Lee, R. M. Guidash, and P. P. Lee, "Partially pinned photodiode for solid-state image sensors," U.S. Patent 5,903,021, Jan. 1997.

[2.19] News article by Joshua Fruhlinger: IBM cools stacked silicon chips with water

[2.20] Satish G. Kandlikar: Review and Projections of Integrated Cooling Systems for Three-Dimensional Integrated Circuits

[2.21] Dave Litwiller, DALSA: CCD vs. CMOS Facts and Fiction Photonics Spectra (January 2010)

[2.22] Lattice Semiconductor: Sub-LVDS-to-Parallel Sensor Interface Bridging

[2.23] John Goldie - Manager of Interface Applications: The Many Flavors of LVDS

[2.24] MIPI Alliance, Inc.: DRAFT MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2)

[2.25] Li, X.; Gunturk, B.; Zhang, L. Image Demosaicing: A Systematic Survey. In Proc. SPIE, volume 822, 68221J (2008)

[2.26] Robert A. Maschal Jr., S. Susan Young, Joe Reynolds, Keith Krapels, Jonathan Fanning, and Ted Corbin: Review of Bayer Pattern Color Filter Array (CFA) Demosaicing with New Quality Assessment Algorithms http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA513752

[2.27] David Alleysson, Sabine Süsstrunk, Member, IEEE and Jeanny Hérault: Linear Demosaicing inspired by the Human Visual System

[2.28] B. E. Bayer, Color imaging array, US Patent No. 3971065.

[2.29] Katerina Trilyraki: Development of an advanced platform for controlling the imaging parameters of CMOS image sensors (June 2014)

[2.30] High-Quality Linear interpolation for Demosaicing of Bayer-Pattern color images

http://research.microsoft.com/en-us/um/people/lhe/papers/icassp04.demosaicing.pdf

[2.31] Henrique S. Malvar, Li-wei He, and Ross Cutler: Pascal Getreuer: Image Demosaicking with Contour Stencils http://www.ipol.im/pub/art/2012/g-dwcs/?utm_source=doi

[2.32] Gary Sullivan and Stephen Estrop, Microsoft Corporation: Recommended 8-Bit YUV Formats for Video Rendering (April 2002, updated November 2008). https://msdn.microsoft.com/en-us/library/windows/desktop/bb530104%28v=vs.85%29.aspx

[2.33] Clarke, C.K.P., 1986, Colour Encoding and Decoding Techniques for Line-Locked Sampled PAL and NTSC Television Signals, BBC Research Department Report BBCRD1986/2.

[2.34] Mei Yan, Xiwei Huanga, Qixiang Jiaa, Revanth Nadipallia, Tongxi Wanga, Yang Shanga, Hao Yua , Minkyu Jeb, Kiatseng Yeoa: High-speed CMOS Image Sensor for High-throughput Lensless Microfluidic Imaging System

[2.35] Chantzi Efthymia: Spectral Cube Construction From Hyper Spectral Scanning Imaging, Chapter 7

[2.36] Norman Koren: Introduction to modulation transfer function (MTF) http://www.normankoren.com/Tutorials/MTF.html

[2.37] Willy Wriggers, Ph.D.: Human Vision and Color

[3.1] Katerina Trilyraki: Development of an advanced platform for controlling the imaging parameters of CMOS image sensors (June 2014)

[3.2] Texas Instruments Wiki: "L130/C6748 Development Kit (LCDK)" http://processors.wiki.ti.com/index.php/L138/C6748_Development_Kit_%28LCDK%29

[3.3] Thomas Lundmark, Lulea University of Technology, Sweden: Development of a USB camera Using a General Purpose Microcontroller http://epubl.ltu.se/1402-1617/2009/058/LTU-EX-09058-SE.pdf

[4.1] Liao Bruce: Intel ® Atom Processor E3800 Series: MIPI * CSI-2 * Camera Sub System

[4.2] LI-OV4689-MIPI Data Sheet

[4.3] Honglu Hou, Wenfang Zhang, Dingjin Huang, and Tao Zhang: Design and Realization of Real-Time Image Acquisition and Display System Based on FPGA

[4.4] Yahia Said, Taoufik saidani, Fethi Smach, Mohamed Atri, Hichem Snoussi: An FPGA-based Smart Camera System

[4.5] Daniel Crispell, Brown University: Implementation of a Streaming Camera using an FPGA and CMOS Image Sensor

[4.6] Xuemei Guo and Tao Liu, Xinxiang University, Henan Xinxiang, China: Application of FPGA in high-speed CMOS digital image acquisition and color recognition system

[4.7] Likun Tian, 2 Xiaohong Liu, 3 Jie Li, 4 Xiaoguang Guo: Image Preprocessing of CMOS Image Acquisition System Based on FPGA

[4.8] Ignacio Bravo *, Javier Baliñas, Alfredo Gardel, José L. Lázaro, Felipe Espinosa and Jorge Garcí: Efficient Smart CMOS Camera Based on FPGAs Oriented to Embedded Image Processing

[4.9] DS150 (v2.5) August 20, 2015: Virtex-6 Family Overview

[4.10] NILADRI ROY and TED MARENA: Using FPGAs for camera sensor interfaces

[4.11] National Instruments: Common Terminology and Definitions for Digital I/O and Logic Analyzers

[4.12] Lambda Photometrics Whitepaper: CoaXPress – High speed camera interface http://www.ukiva.org/pdf/pr/2012/CoaXPress-HSCI_white_paper.pdf

[4.13] Dalsa: HSLink Whitepaper http://www.adept.net.au/news/newsletter/201003-mar/Resources/HSLink.pdf

[4.14] Max Robertson, Analog Applications Engineer, Texas Instruments, Mike Miethig, Chair, CameraLink HS, and Technical Manager, Teledyne DALSA: Implementing a CameraLink HS Interface Using the TLK3134 http://www.ti.com/lit/wp/slaa533/slaa533.pdf

[4.15] ©2014 Eqcologic NV: Engineering Information for EQCO62X20 chipset

[4.16] Point Grey White Paper Series: USB 3.0: Improvements over USB 2.0 https://www.ptgrey.com/white-paper/id/10692

[5.1] CYUSB303X: EZ-USB® FX3S SuperSpeed USB Controller http://www.cypress.com/file/139246/download

[5.2] CYUSB3KIT-003 Doc. # 001-93186 Rev. *A : SuperSpeed Explorer Kit User Guide

[5.3] AN75779 - How to Implement an Image Sensor Interface with EZ-USB® FX3™ in a USB Video Class (UVC) Framework http://www.cypress.com/documentation/application-notes/an75779-how-implement-image-sensor-interface-ez-usb-fx3-usb-video

[5.4] USB Implementers Forum: Universal Serial Bus Device Class Definition for Video Devices Revision 1.1 June 1, 2005

[5.5] Aptina Imaging: MT9M034: 1/3-Inch CMOS Digital Image Sensor Features Rev. E (2010)

[5.6] Aptina Imaging: MT9M024 and MT9M034: Register Reference Register Descriptions Rev. B (2011)

[5.7] http://www.mouser.com/ds/2/276/0525593652_FFC_FPC_CONNECTORS-170634.pdf

[5.8] Cypress Semiconductor : EZ-USB® FX3™ Technical Reference Manual

[5.9] Rizwan Afridi, Hussein Osman: EZ-USB® FX3™/FX3S™ Hardware Design Guidelines and Schematic Checklist (page 8).

[6.1] Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing (3rd Edition)

[6.2] Willy Wriggers, Ph.D., School of Health Information Sciences: Human Vision and Color http://biomachina.org/courses/processing/12.html