Development of External Readout Circuitry for Digital X-ray Imager

by

Rasoul Keshavarzi-Valdani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada 2012

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

X-ray detectors are structurally very similar to visible light detectors. They transform the high-energy photons in X-ray beam into an electrical signal, either a voltage or current, that is detected by means of an electronic circuit. Single pixel X-ray detectors are put together in a 2D layout to construct a panel of X-ray detectors. Once an X-ray panel detector is exposed to X-ray beam passing through the object under study, an image is formed on the panel.

In this work an interface is designed, built, and tested that reads out the pattern generated by a 32x32 flat panel X-ray detector. There are two main sections in any X-ray panel detector, Gate Drivers and Charge Amplifiers. In order to minimize the size of the circuit, which can be very large if discrete components are used for both sections, an application specific integrated circuit (ASIC) designed for X-ray panel detection purposes is employed. The AISC, Gate Drivers, and other necessary electronic components are mounted on a main-board that hosts the X-ray panel detector as well.

A field gate programmable array (FPGA) connects the main-board and a personal computer (PC), and has been programmed to serve multiple purposes; it generates the pulses required to drive the X-ray pixel sensors on the panel, receives the fast stream of data coming from the ASIC, and transfers the information to the PC using RS-232 protocol.

A program running on the PC receives the data sent by FPGA over the serial cable, and puts them together to regenerate the image captured by the X-ray panel detector on the monitor of the computer.

In order to test the functionality of the overall system, a 32x32 passive pixel sensor (PPS) with lateral amorphous Selenium metal-semiconductor-metal (MSM) structure was used. The array has been tested under blue light which is the closest visible light to X-ray spectrum, and some captured images are presented. Several techniques for creating better quality images with less noise are also discussed.

Acknowledgements

I would like to sincerely thank my supervisor, Professor James A. Barby, for accepting me as one of his graduate students. This selection has indeed been a turning point in my life, allowing me to go back to school after over ten years of industrial working experience.

I am grateful to Professor Karim S. Karim for agreeing to be my co-supervisor and giving me the opportunity to collaborate with him in the field of medical imaging. It was a truly joyful experience to work with him. Not only did I learn much technical knowledge from him, but more importantly I learned a lot about interpersonal skills from him. His enthusiasm and positive outlook in all aspects of life have made him an exemplary personality for me.

This is also an opportunity to thank Professor Ajoy Opal and Professor Peter Levine for reading my thesis during their busy daily schedules and providing valuable feedback. Their comments caused my thesis to be much more uniform and smooth in terms of communicating with readers.

Thanks to Dr. Kai Wang for his insightful suggestions on testing the prototype system, to Dr. Mohammad Y. Yazdandoost for making a sample imager, to Christos Hristovski for his help in wire-bonding of the sample imager, to Dr. S. Hasibul Majid for depositing Selenium on the sample array, to Dr. Mohammad-Hadi Izadi and Dr. Amirhossein Goldan for their help in PCB design, to Dr. Nader Safavian and Shiva Abbaszadeh for their help in VHDL coding and testing of the prototype design, to Nicolas Allec for his help in C programming, and to Laura Winger for her help in making fixtures for the prototype system.

It was a great pleasure to be a member of the STAR group and work with its members, including Umar Shafique, Dr. Michael Adachi, Kyung-Wook Shin, Chris Scott, Sina Ghanbarzadeh, Yuan Fang, and Dr. Fung Chen.

Finally, my deepest gratitude goes to my wife, Nouri, who has always been patient, caring, and supporting. This work is dedicated to my daughter Sara and my son Ryan, knowing that I often had to cut short my time with them in order to work on this research.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

This chapter gives a brief description of X-rays and how they are used in medical systems. It starts with a quick review of X-ray discovery. Then, a unique feature of X-rays that has made it useful for medical applications is discussed. X-ray generation is described in short. Finally, the X-ray detection is reviewed in more detail describing why X-ray detection has been under research since the invention of X-ray beam.

1.1 Discovery of X-ray

The X-ray beam was discovered accidentally by a German scientist called Wilhelm Conrad Röntgen in 1895. It is a spectrum of electromagnetic wave which falls in a certain frequency band, as shown on Figure 1.1 [1]. Visible light spectrum is also given on Figure 1.1 for comparison.

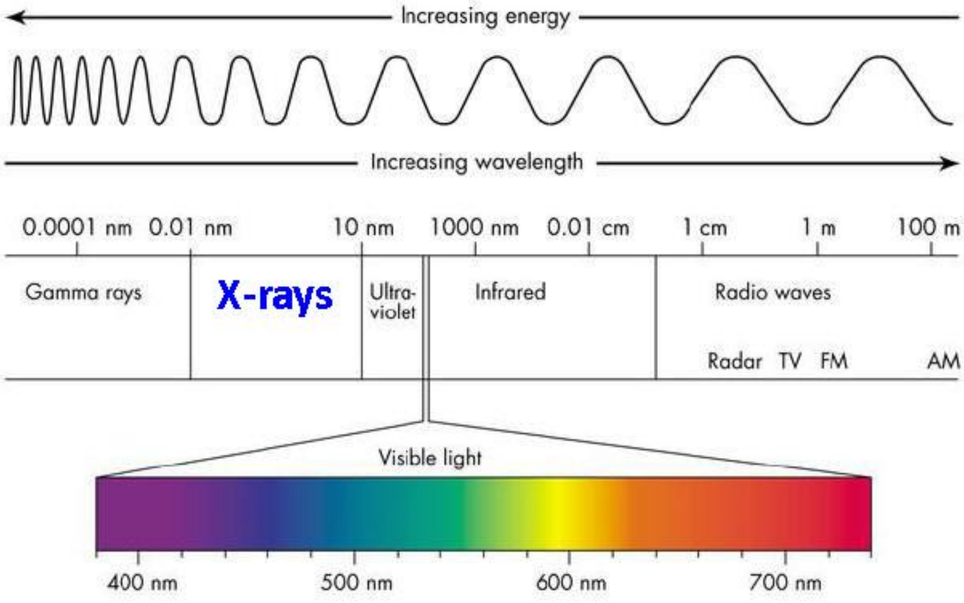


Figure 1.1 – Electromagnetic waves spectrum [1]

Similar to visible light, X-ray beam is also made of photons and therefore belongs to the Electromagnetic waves. In general, waves with wavelength between 0.01 – 10 nm is considered as X-ray band.

1.2 Material properties of X-ray

When a material is exposed to X-ray, similar to when the material is exposed to visible light, different scenarios can happen. Since the outcome of this interaction depends on the energy of the photon, let us limit the discussion to a fixed energy photon.

When an X-ray beam of photons with fixed wavelength (i.e. energy) collides with a material, two different reactions can be observed:

- The material can be transparent to the photons with that energy level, like a piece of glass that is mostly transparent against visible light spectrum. In this case most of the photons in the beam pass through the material.
- The material can attenuate the incident beam of X-ray photons. The process of attenuation depends on the interaction of the photons with atoms and electrons in the material. The photon energy can be completely absorbed to generate an electron-hole pair. Or the energy of the colliding photon can partially be used to excite a valence electron to escape from the nucleus field, resulting in a photon with less energy plus an ionized atom in the material.

This is the main property of X-ray beam, that it passes through some material while attenuated by other types of materials. In fact, the ratio of attenuation to penetration is proportional to material density. It means, materials with higher density will cause more attenuation compared to lower density materials.

The first picture ever taken using X-ray beams was Röntgen’s wife’s hand shown in Figure 2.1 [2].

1.3 X-ray generation

In section 1.2 the attenuation as a possible result of collision between a beam of photons and a material was described. The X-ray generation historically was inspired from that phenomenon.

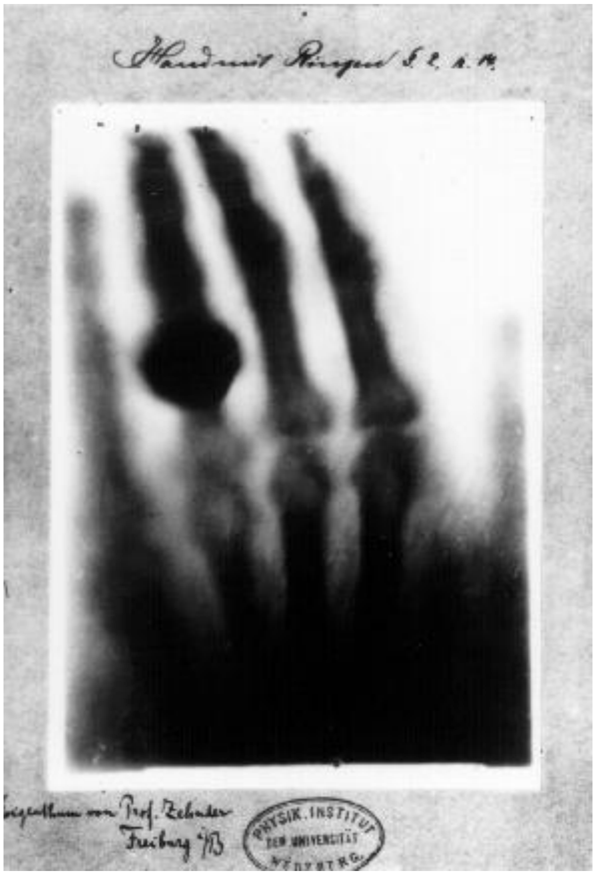

Hand mit Ringen
igenthum von Prof. Zehnder
Freiburg i/B
PHYSIK. INSTITUT DER UNIVERSITÄT WÜRZBURG

Figure 1.2 – First picture ever taken using X-ray by Röntgen [2]

If a beam of high energy electrons (not photons) is directed towards a material (target), some colliding electrons might interact with the atoms and electrons in the target. Depending on the energy of colliding electrons and properties of the material under bombardment, some incoming electrons might release some energy to the electrons that belong to atoms of the material. A photon is generated in many cases in order to satisfy the law of energy conservation. If the energy of the incidental electron and the material type are chosen properly, many of these generated photons can belong to X-ray band. Figure 3.1 shows how this is done in an X-ray tube [3].

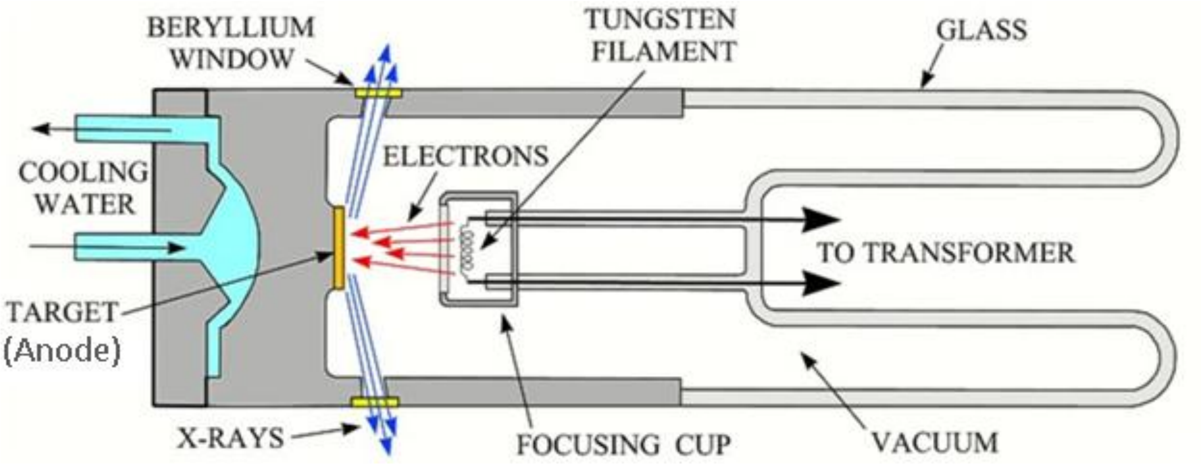


Figure 1.3 – Schematic cross-section of an X-ray tube [3]

The Tungsten filament generates free electrons, similar to what is done in Cathode Ray Tubes. A very high voltage difference (around 40 kV) between the anode and cathode, gives enough energy to electrons before reaching to target.

Since a century ago to this date commercial X-ray tubes have suffered from low efficiency. It means only a small percent of the consumed energy is transformed into X-ray beam and most of energy is lost as heat generated. This is why in Figure 3.1 there are some pipes for cooling water.

Fortunately new research has shown that there are ways to produce X-ray beam with much higher efficiency using carbon nanotubes [4]. But it is in research phase and no commercial product is available at this time.

1.4 X-ray detection

Once a material is exposed to a beam of X-ray photons, depending on the material type and mass density at different spots, some photons penetrate and pass through the material without attenuation, while some others might be attenuated as described in section 1.2. Figure 4.1 shows how X-ray is used for medical purpose [5].

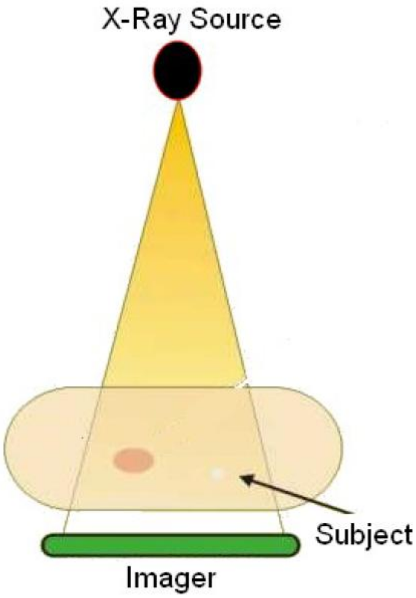


Figure 1.4 – Radiography using X-ray beam [5]

The intensity of the incidental X-ray beam at each spot on the detector plane should be measured in order to capture a two-dimensional picture of the material under experiment. This is where X-ray detectors and its accuracy becomes important.

X-ray detectors have been under research for the past century. Analog detectors and X-ray films have been used most of this time. Within the past two decades, digital detectors have mainly replaced them in most hospitals in modern countries.

1.4.1 Analog detectors

Light sensors have been around for several decades, well before the digital era. In earlier days, black and white cameras used materials that were sensitive to light intensity. After chemical processing of the film reel, the brightness at each spot was proportional to the light intensity of the incident light to that spot and the duration the spot was exposed to that incoming light. X-ray detectors have traditionally made products similar to those of the black and white film industry; the output was black and white slides taken of the object under the X-ray beam. The grey level at each spot depends on three factors:

- Intensity of the incident X-ray beam at the spot
- Duration of X-ray bombardment to the spot
- Hardness (absorption) of the object through the path of the X-ray beam at the spot

It is the last factor that the whole medical X-ray industry is based on. In other words, an X-ray image produces a shadow of the interior of the object under the beam. Dark areas mean that the majority of the incident X-ray beam has reached the detector, which means the object was mostly transparent against X-ray for those areas. Bright areas mean that the object has blocked (absorbed or reflected) the majority of the incident X-ray beam. Figure 5.1 shows a contemporary X-ray image of a hand, similar to Figure 1.2 [6].

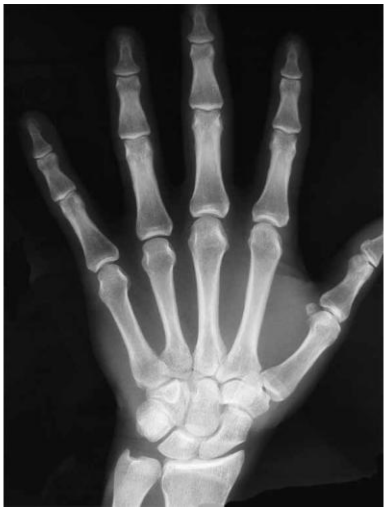


Figure 1.5 – Contemporary X-ray image of a hand [6]

In Figure 1.4, since bones have denser tissue compared to the rest of the hand, they are shown in the picture as light areas.

For visible light detection and storage systems, there was huge demand to move to full color outputs, as opposed to black and white. Thus, the whole camera and movie industry is now using color detectors. However for X-ray detectors, the output has remained black and white to this date. In fact there have been some efforts to make color X-ray results, but that trend was not accepted by the majority of doctors and medical staff, the prime users of medical X-ray products. Therefore, an X-ray image is still expected to show a black & white picture of the object under X-ray beam.

1.4.2 Digital detectors

Similar to the visible light detector industry that has adopted digital detection, storage, and retrieval systems within the past decade or so, the X-ray industry has also shifted towards digital systems slowly in the same time span. A digital X-ray system has many advantages as opposed to conventional analog X-ray film system including:

- Faster response simply because there is no need for chemical processing
- Quality of captured pictures would not degrade with time.
- Easier to electronically transfer results from one spot to another
- Ability to zoom into suspicious areas

For a digital radiography system, the imager shown in Figure 1.4 is a digital panel as opposed to analog film layer.

Digital detectors are divided into two main streams: Direct detectors and Indirect detectors.

1.4.2.1 Direct detectors

In this method a thick photoconductor material is used as a medium that generates electron-hole pairs when a photon with energy level in the X-ray band hits an atom. A strong electric field across the photoconductor material separated the electron and hole and causes them to move in opposite directions. Figure 1.6 shows a cross section of such type of conversion [7].

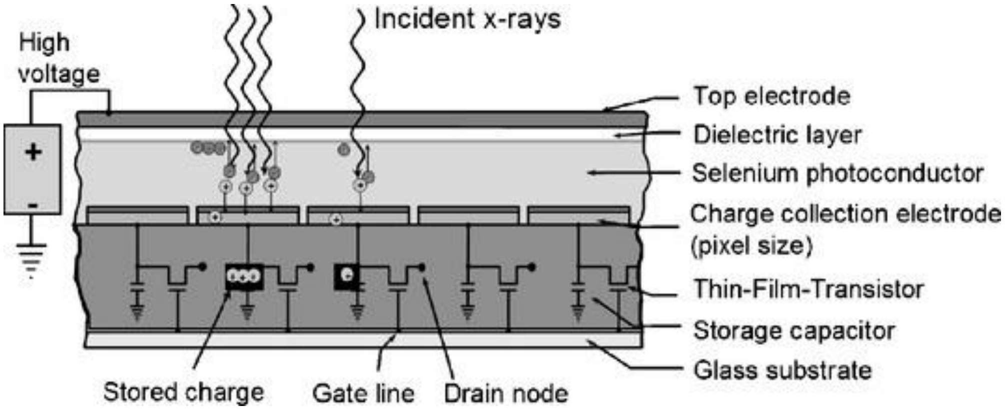


Figure 1.6 – Direct detection of X-ray [7]

The resulting current which is a sum of electron and hole currents is then measured by the underlying electronics. The density of the generated current is proportional to the intensity of the X-ray beam.

1.4.2.2 Indirect method

In this method a scintillator layer is deposited on top of a photodetector layer as shown in Figure 1.7 [7]. The scintillator layer absorbs the X-ray photons and generates some photons with different energy level, like visible or UV, that is then detected by the photodetector layer.

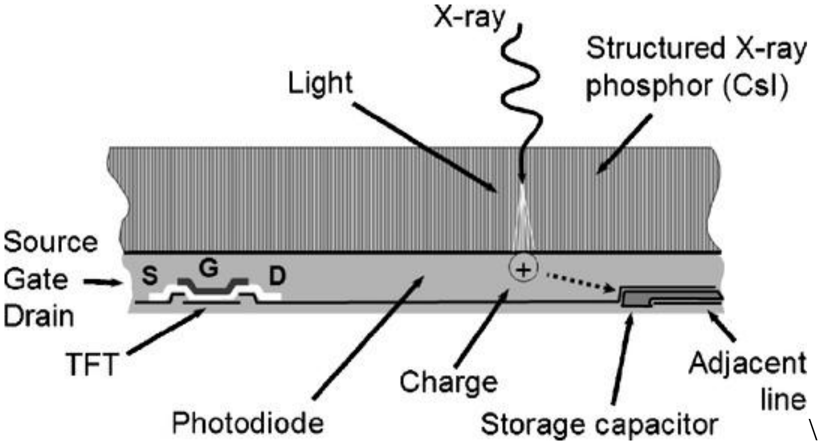


Figure 1.7 – Indirect detection of X-ray [7]

The intensity of the generated visible or UV light is proportional to the X-ray beam intensity. Also the density of the generated hole-pair is proportional to the intensity of the visible or UV light generated by the scintillator layer. Therefore, the intensity of the overall current generated is proportional to the X-ray intensity.

1.4.2.3 Pixel detectors

A digital X-ray panel detector is made of a two dimensional array of single-pixel digital detectors put together in a matrix format. Therefore it is important to know how a single pixel X-ray detector works.

Single pixel X-ray detectors are divided into two types of passive pixel sensor (PPS) and active pixel sensor (APS).

1.4.2.3.1 Passive pixel sensor (PPS)

Figure 1.8 shows a typical PPS circuit. The capacitor represents the sensor, which is charged when the pixel is under X-ray beam.

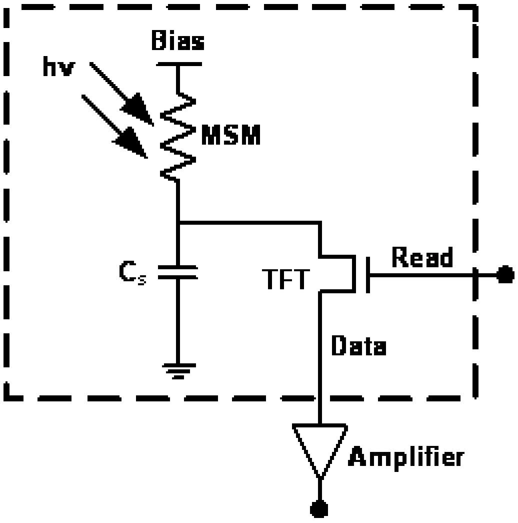


Figure 1.8 – Passive pixel sensor (PPS)

The transistor in this Figure acts like a switch. Once turned on, it allows the voltage across the capacitor to be read by another circuit on the column line.

1.4.2.3.2 Active pixel sensor (APS)

In PPS circuit the detected signal is read by an external readout circuitry without any on-silicon amplification. This is a drawback of this design because the detected signal is very weak and therefore has low SNR (Signal to Noise Ratio).

A second group of detectors employ some sort of on-silicon or on-wafer amplification before sending the signal to external readout circuitry. It is clear that in this case a higher SNR is achievable. Figure 1.9 shows one of this type of detectors called APS (Active Pixel Sensor). There are different circuit types of APS detector and the following one is just a sample.

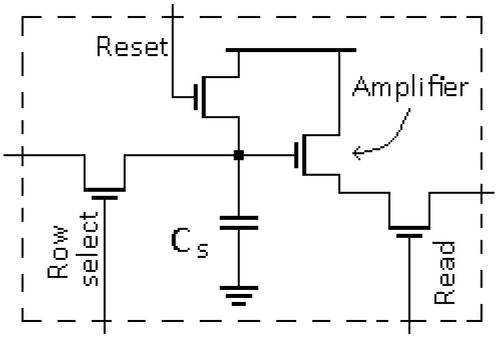


Figure 1.9 – Active pixel sensor (APS)

The APS structure has more transistors on each pixel and requires a more complicated control signal scheme. However it can represent a superior performance when compared to PPS.

1.4.2.3.3 X-ray detector panel

A detector panel, similar to single pixel detectors, can be of PPS or APS type. As the name implies, once the passive pixel sensor detectors are used to construct the panel, then an X-ray panel with PPS detector is built. Figure 1.10 shows the structure of a PPS panel.

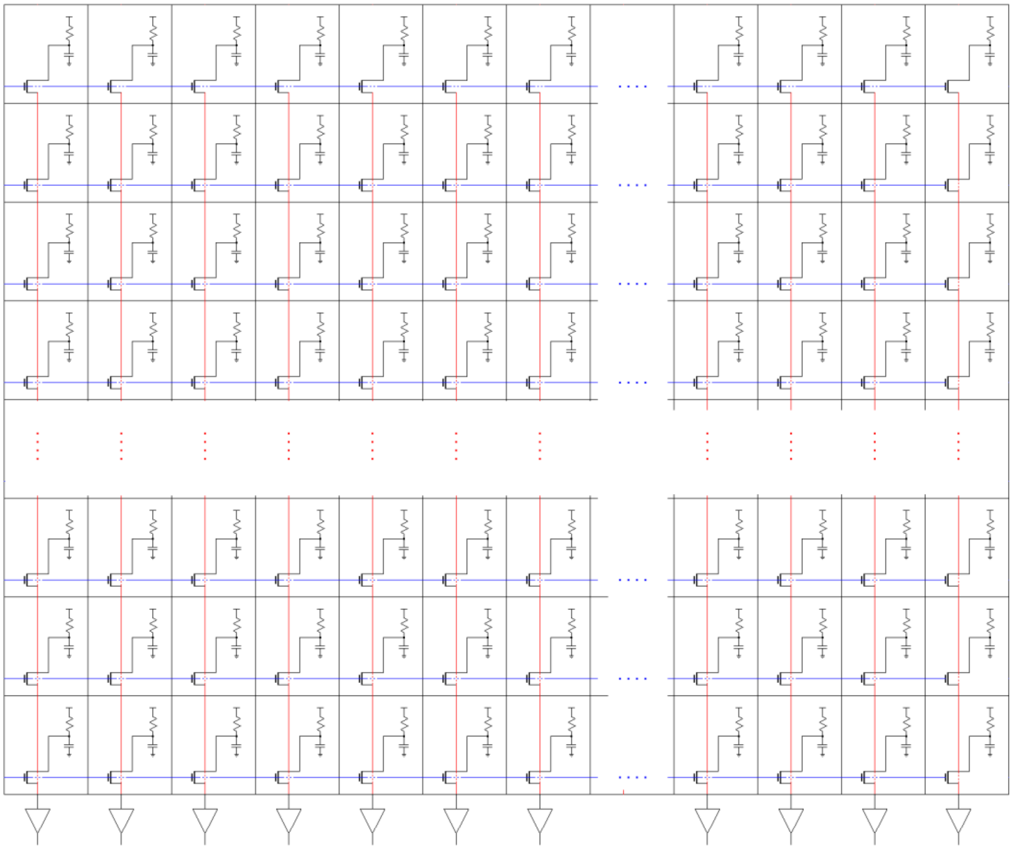


Figure 1.10 – X-ray imager (panel) with PPS detectors

Similarly, an APS panel consists of a two dimensional matrix of single pixel active sensor detectors. Figure 1.11 shows an APS panel. In this Figure there is one additional transistor for each pixel compare to Figure 1.9. It is added to facilitate matrix operation which row and column operation.

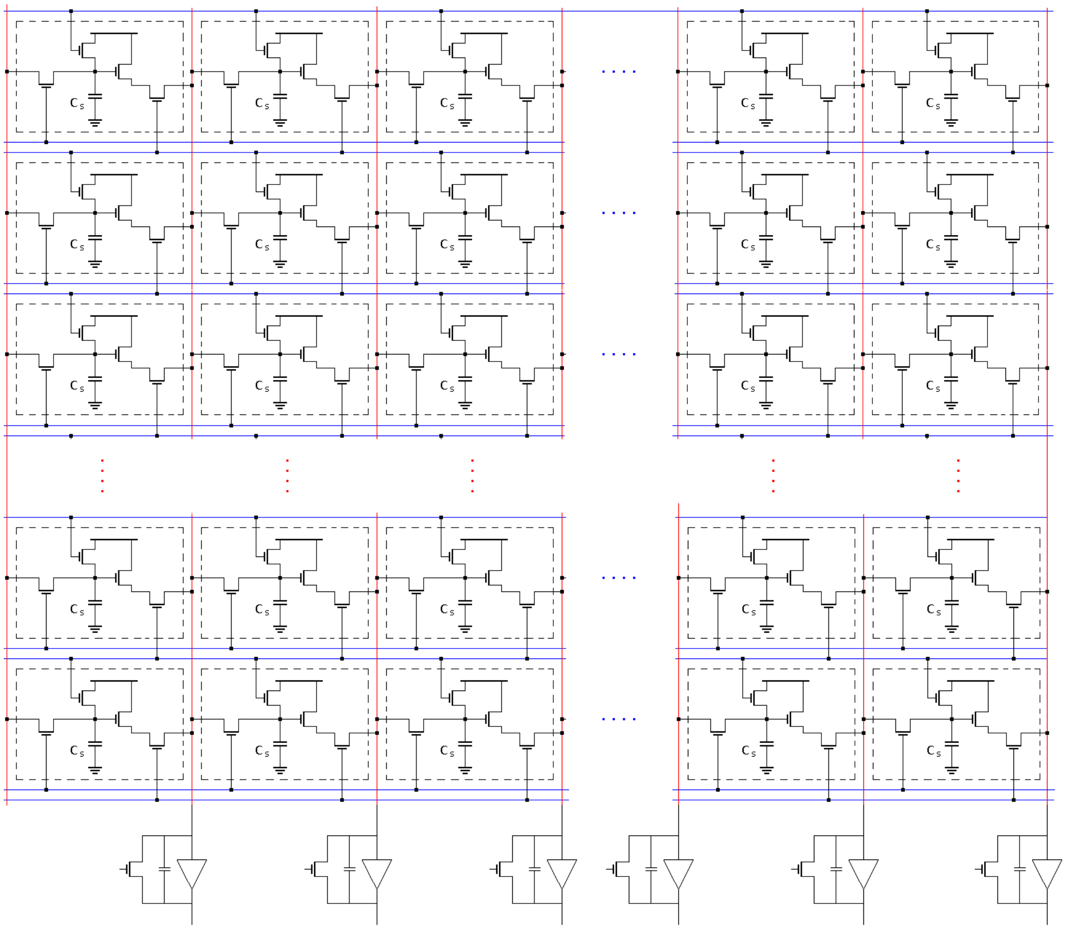


Figure 1.11 – X-ray imager (panel) with APS detectors

1.4.2.3.4 X-ray imager (panel) readout circuitry

Once an X-ray panel is fabricated in the lab, the next step is to extract a frame captured by the panel. Figure 1.12 shows a block diagram of such readout circuitry for an array of size 64x64.

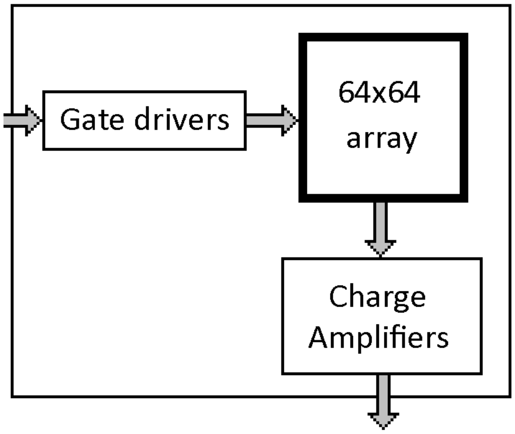


Figure 1.12 – Block diagram of a readout circuitry

The panel itself can be bases on PPS or APS detectors. In any case the output is a series of lines (columns) that should be read by an array of charge amplifiers. The next chapter describes the details of the hardware interface.

Chapter 2 Design of the hardware interface

This chapter starts with a description of the ultimate goal of this work. Then it covers the steps taken to achieve this goal. Failed steps taken are included as well, and the reasons for these failures are clarified in each case. Eventually, a substitute approach to the task is presented.

2.1 Ultimate goal of this work

The goal at its most basic is to build an X-ray panel imager readout circuitry of the maximum practicable size. Having such an interface in hand will give the STAR group members the ability to test their manufactured arrays in-house, which means on-campus. In the past, some members had to take their arrays to off-campus sites in order to test their array and extract sample images produced by their array, a situation that was both expensive and time consuming. The ultimate preferred size for the array reader would be 128 pixels by 128 pixels, a size determined by the components/devices used for the readout circuitry. The factors that influenced this choice will be clarified later.

2.2 Design challenges

An X-ray panel imager is nothing more than a matrix of single pixel X-ray detectors. Hence the first solution that may come to mind is just repeating the same single pixel detector circuit for a matrix with a size of m by n. This simple solution, unfortunately, is not easily feasible, especially for large matrix sizes. An X-ray panel imager readout interface has two modules, “Gate Drivers” and “Charge Amplifiers”, as the main blocks. Each block can grow drastically in size if one wants to use discrete components in designing such circuits. For example, Figure 2.1 shows a sample single pixel X-ray detector circuit often used by STAR group members to measure their designs’ performance.

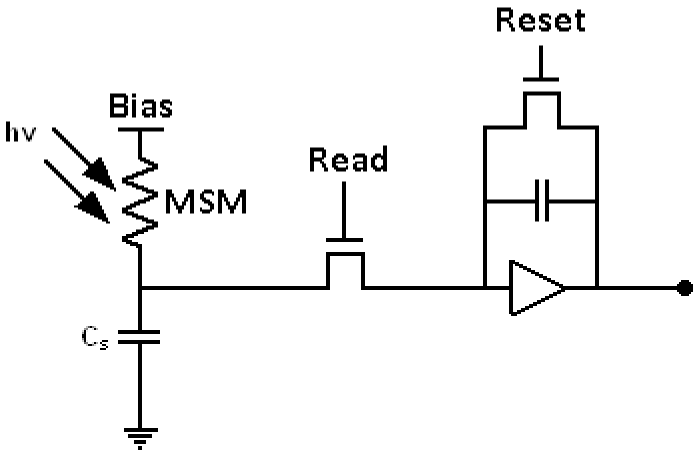


Figure 2.1 – Sample circuit for single pixel X-ray detector

In this circuit, an integrated circuit with part number of IVC102 is used as charge amplifier. This device comes in different packages, with SO-14 as the smallest one. It contains two charge amplifiers. A similar IC in the same family has four charge amplifiers in one package. With the goal of designing a readout circuitry of 128x128 size, 32 of similar type of IC with quad output should be employed for the charge amplifier section. Similarly, a large circuit has to be employed for the gate driver section. From PCB design point of view, populating such a large number of ICs on an oversized board, and routing power/ground/signals to these ICs is tedious. Moreover, from signal integrity point of view, such a design has some intrinsic flaws. For example, the length of lines in the charge amplifier section will inevitably be unequal. This means parasitic capacitance associated with such lines varies across the array, and consequently, there is non-uniform noise bias across the array.

To eliminate oversized PCB issues, it is better to avoid using discrete components and instead, integrate the circuit as much as possible. A search for integrated gate drivers and charge amplifiers brings different options such as ASTLC5302A for gate driver, and ISC9717 or TXR5801A for charge amplifier blocks. Since these devices are manufactured for industrial use, access to small quantity of them for research purpose is not always possible. For reasons beyond the scope of this work, it was already decided to use the ISC9717 chip manufactured by FLIR as a charge amplifier. The ISC9717, also known as the Indigo chip, has 128 input pins with an integrated charge amplifier on each line. In consequence, the ultimate size of the detector is 128 pixels by 128 pixels, which can be achieved by using only one Indigo chip. Larger sizes will need more than one Indigo chip, which just adds non-necessary complication to the task. So, the ultimate goal is to build an X-ray readout circuitry for arrays up to 128x128 pixels using one Indigo chip.

The Indigo chip comes in a Silicon die format. In other words, it comes unpackaged. Thus, the first concern is, how to integrate this piece of Silicon die into the interface.

2.2.1 Silicon die on PCB

The first approach tried was the die-on-PCB method. In this way the Silicon die is glued on a PCB in a place especially designed for it. Then wire-bonds connect bond-pads on the Silicon die to corresponding bond-pads on the PCB. Figure 2.2 shows the top view of the Indigo die [8]. The Indigo die size is 11.15 mm by 4.74 mm.

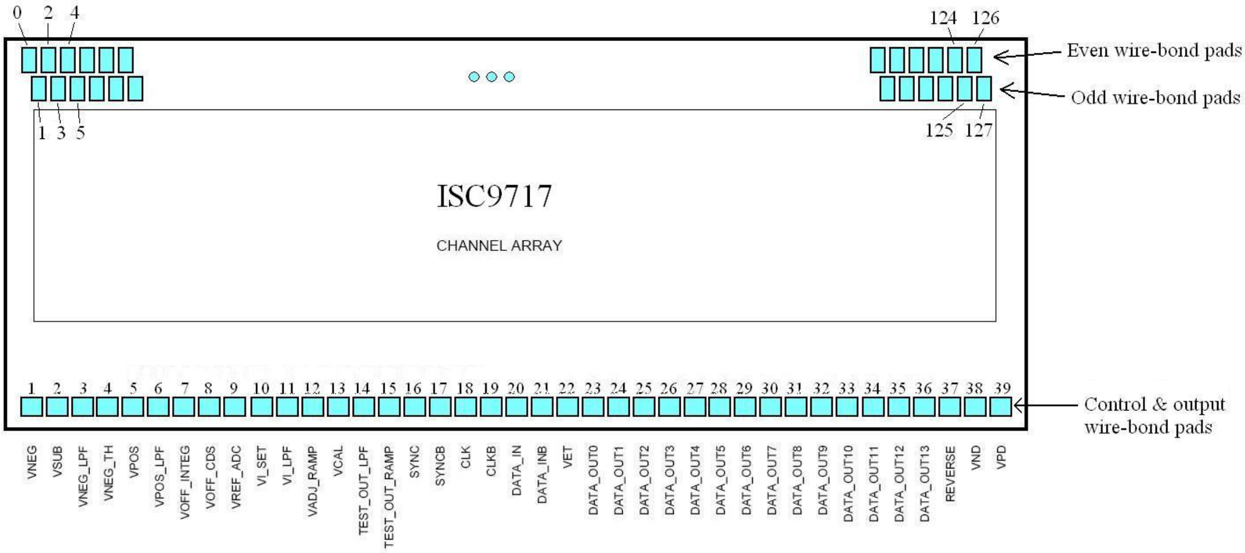


Figure 2.2 – Indigo Silicon die top view [8]

The first solution that may seem easiest is creating a wire-bond pattern on the PCB similar to the wire-bond pattern on the Silicon die. This idea is shown on the Figure 2.3.

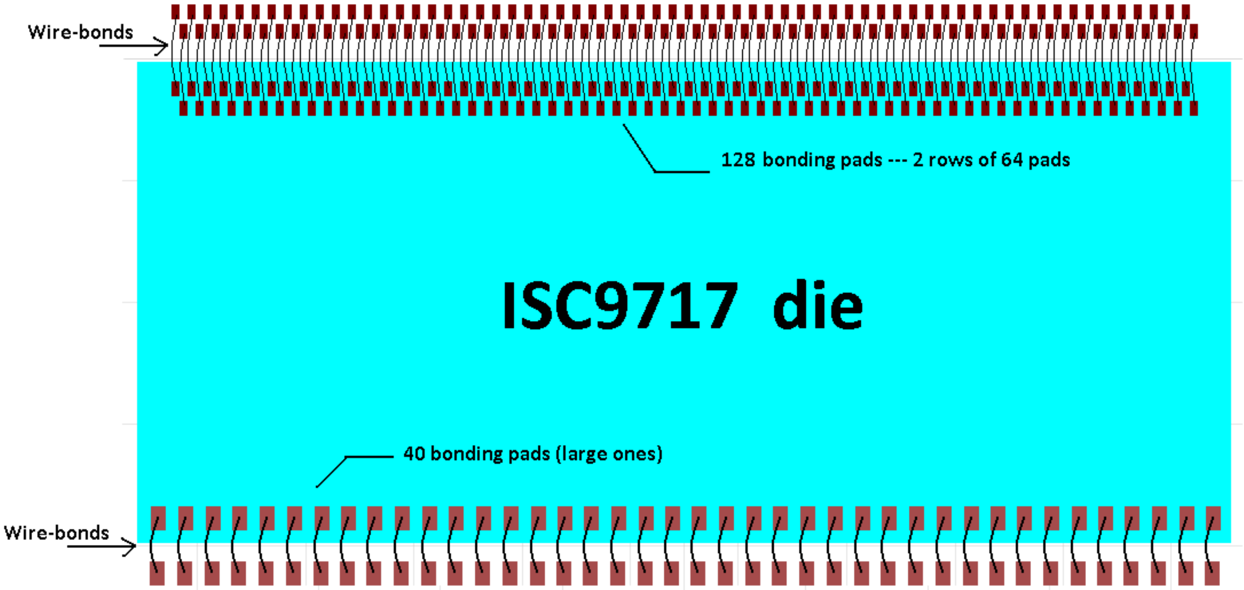

Wire-bonds
128 bonding pads --- 2 rows of 64 pads
ISC9717 die
40 bonding pads (large ones)
Wire-bonds

Figure 2.3 – Can the Indigo Silicon die be wire-bonded to PCB like this?

But the pattern shown in Figure 2.3 is not easily implementable on PCB. The reason is the small feature sizes of this pattern. For example, the bond-pads on the side with two parallel rows are 80 micron by 160 micron as shown in Figure 2.4 [8]. The distance between adjacent bond-pads is also 80 micron. This 80 micron pitch is a huge size on semiconductor processing side on the Indigo chip. But it is too aggressive when it comes to do it on PCB manufacturing. In other words, with the current technology, manufacturing of a PCB with such sizes is quite possible, but since it is small quantity order with features more challenging than the regular/normal specifications, the order will have to be considered as custom order, which results in very high prices.

Here are the important design rules used for PCB manufacturing:

Minimum copper width	7 mil
Minimum drill diameter	10 mil
Clearance (all)	7 mil

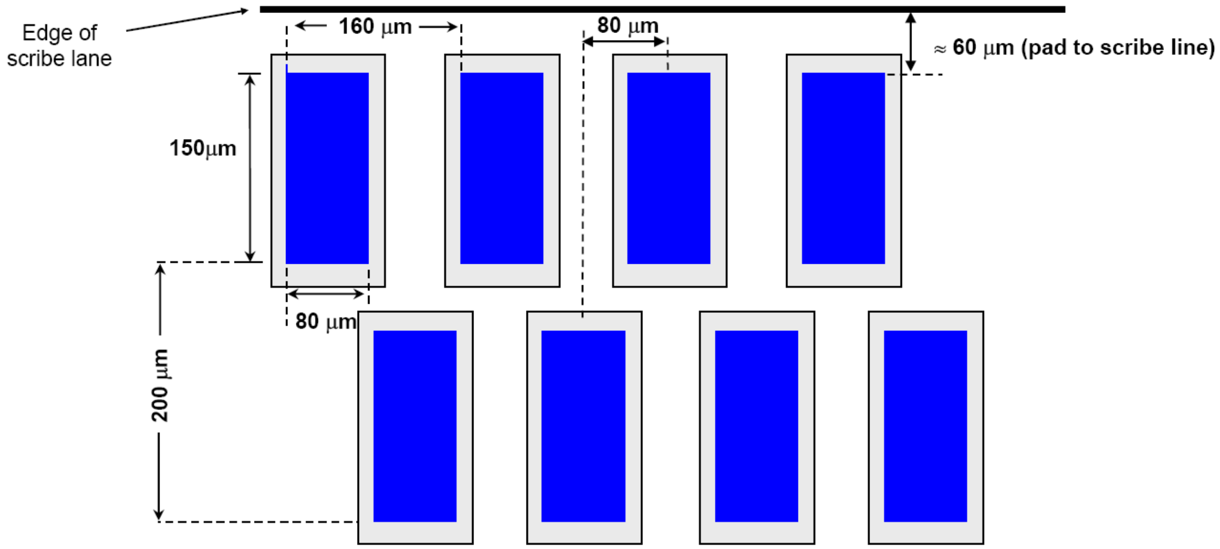


Figure 2.4 – Bond-pads on the inputs side on the Indigo chip [8]

One solution to go around this issue is putting several rows of bond-pads in parallel. Figure 2.5 shows this method. Even though the issue of small feature sizes on PCB no longer exists, but there are connections with long wire-bonds in this new topology. Having long wire-bonds itself is not a problem. But having long wire-bonds very close to each other makes the wire-bonding task difficult from one side, and makes the product very susceptible to shake. In other words, a small movement will be enough to cause two adjacent wire-bonds to touch each other.

Considering all above facts, it was concluded that the die-on-PCB approach will not provide a robust solution. So, this approach was discarded and other solutions tried out.

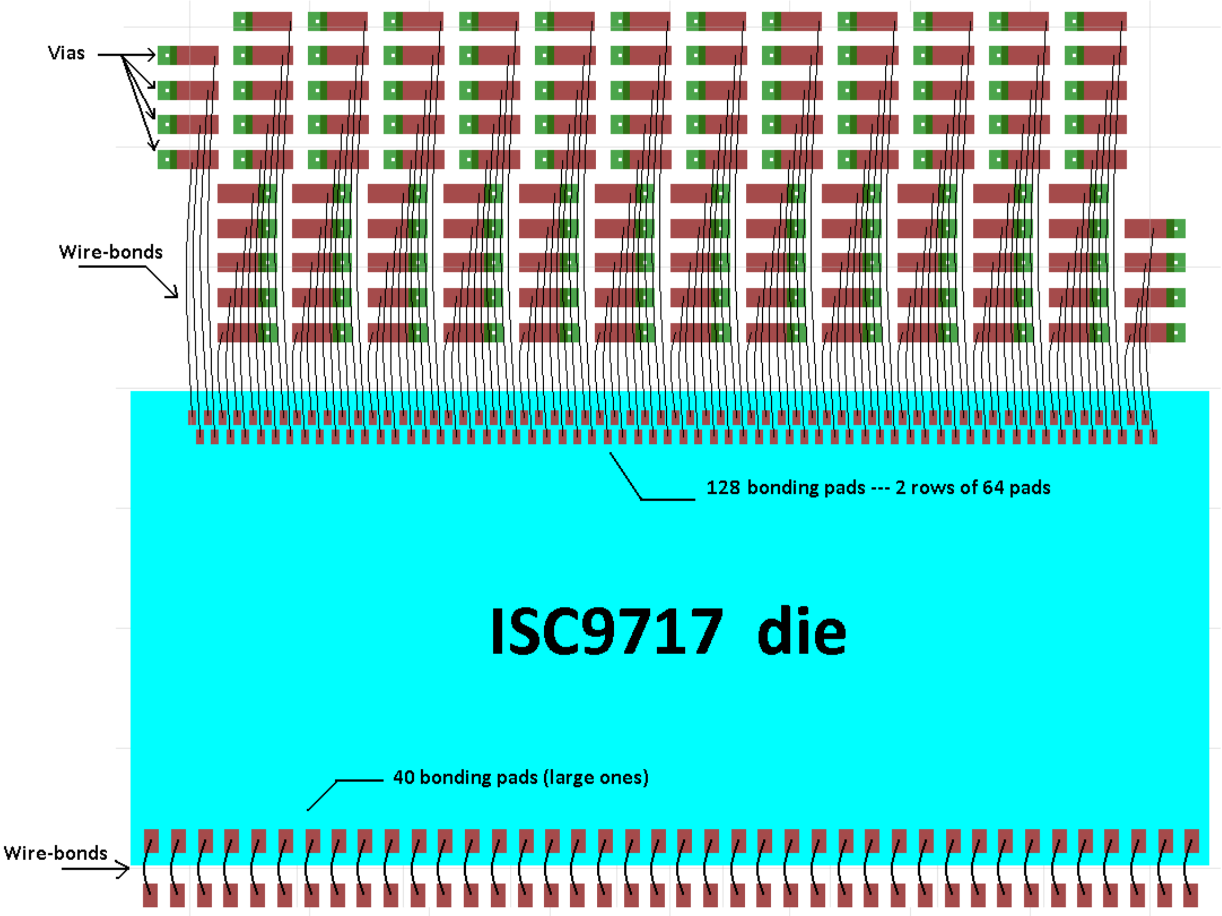

Vias
Wire-bonds
128 bonding pads --- 2 rows of 64 pads
ISC9717 die
40 bonding pads (large ones)
Wire-bonds

Figure 2.5 – A solution for too small sizes on PCB

2.2.2 Package for Indigo

Figure 2.6 shows a package and how a semiconductor-die wire-bonded to it. In today’s technology market it is quite easy to find a package with hundreds of bond-pads and pins. But when the number of bond-pads/pins increases, manufacturers would have to populate the package’s cavity with bond-pads on all four sides of the cavity.

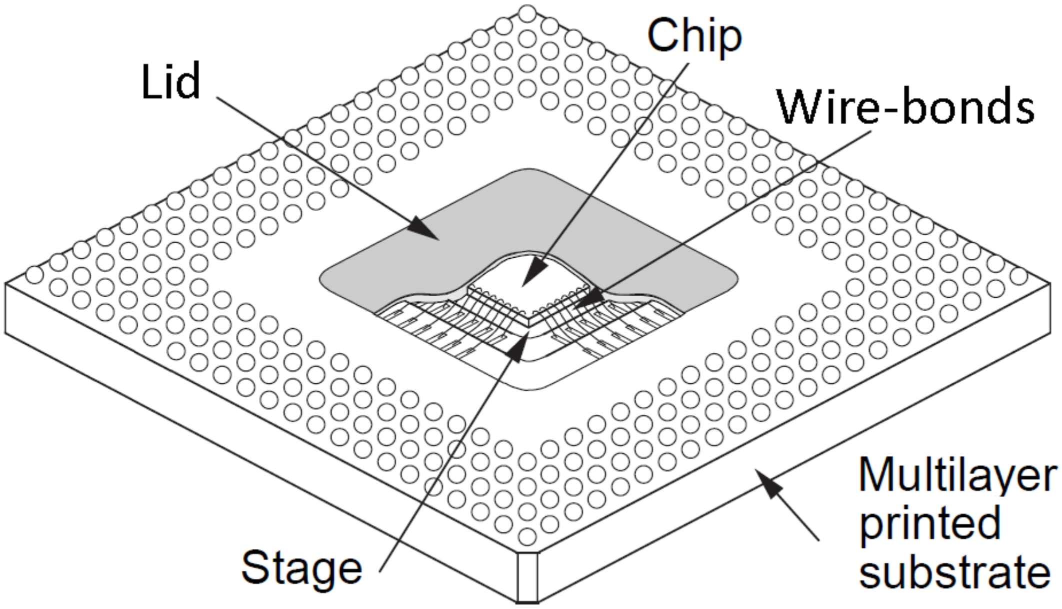


Figure 2.6 – A package and a semiconductor-die wire-bonded to it

In the case of Indigo chip, the bond-pads are on two sides of the Silicon die only. This makes it difficult to find a package that provides 128 separate bond-pads/pins on one side only.

Thanks to CMC (Canadian Microelectronic Corporation), the CPG44705 package came to our attention that encompasses 447 pins altogether. But out of these pins, there are a lot of them used for VCC (positive supply voltage) and VSS (negative supply voltage – often the same as Ground). Figure 2.7 shows the layout of the package's cavity.

The maximum number of independent (not shorted to VCC or VSS) bond-pad/pins on one side of the cavity is 100. This is less than the 128 that is our ultimate goal. But the difference is not that much either. So, if the wire-bonder technician is experienced enough to connect 14 more wire-bonds on each side, then all 128 bon-pads would be connected to outside pins of the package. A sample wire-bond diagram for such case is shown in Figure 2.8.

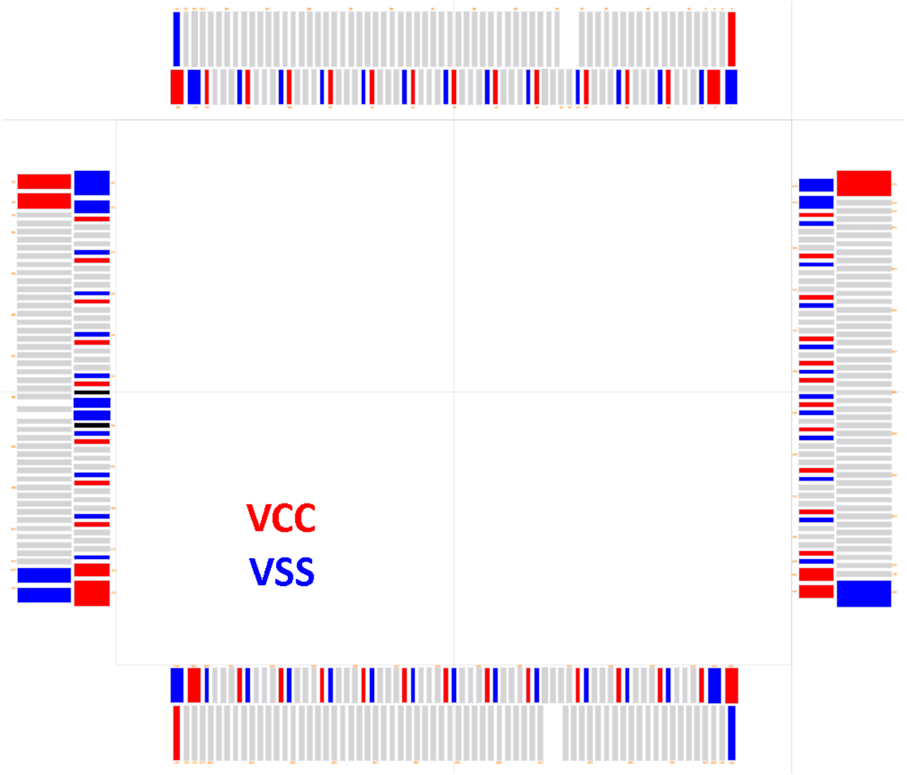
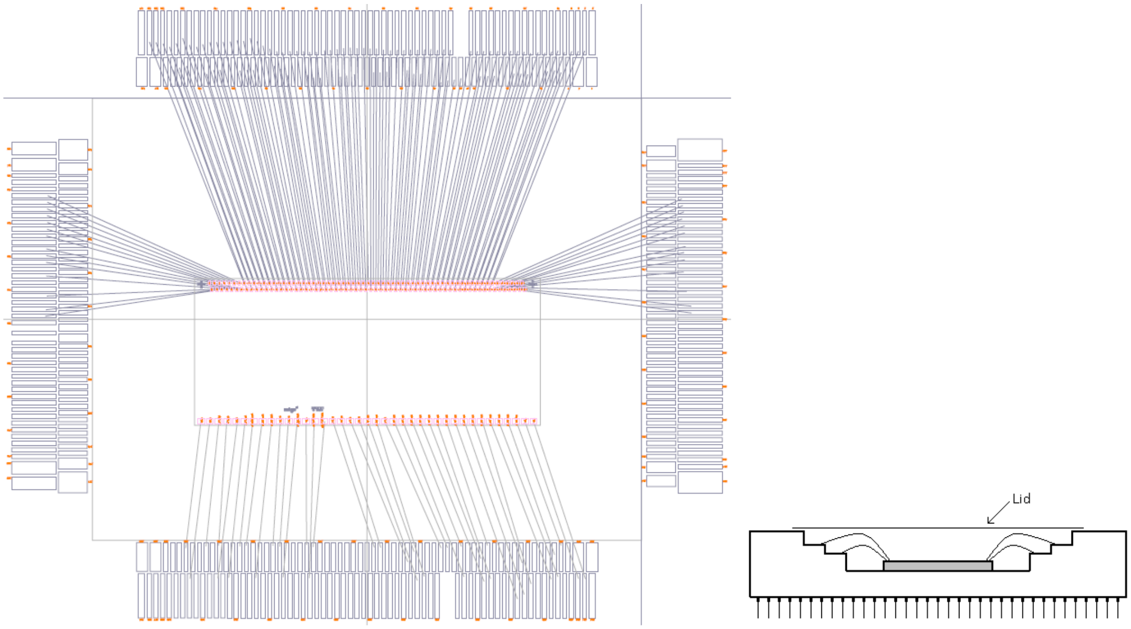


Figure 2.7 – Layout of the CPG44705 package’s cavity



(a) 14 wire-bonds connected to each side

(b) Cross section of the package

Figure 2.8 – All inputs of Indigo wire-bonded to bond-pads

In spite of the fact that this diagram seems feasible, we realized it was difficult to find a contractor that would accept the challenge, especially if it is for low quantity, not mass production. In other words, the cost would be beyond acceptable range for this project.

After e-mail correspondence with some wire-bonding facilities, it was concluded that in order to speed-up this task, some requirements have to be slashed to make the task easier from manufacturing point of view. So, a second wire-bond diagram consisting of only 64 wire-bonds to input pins, as shown in Figure 2.9 was created and sent to a contractor. This reduction in turn limited the maximum number of columns to 64 for our interface. In other words, our ultimate goal from now on will be building an interface for arrays up to 64x64 pixels.

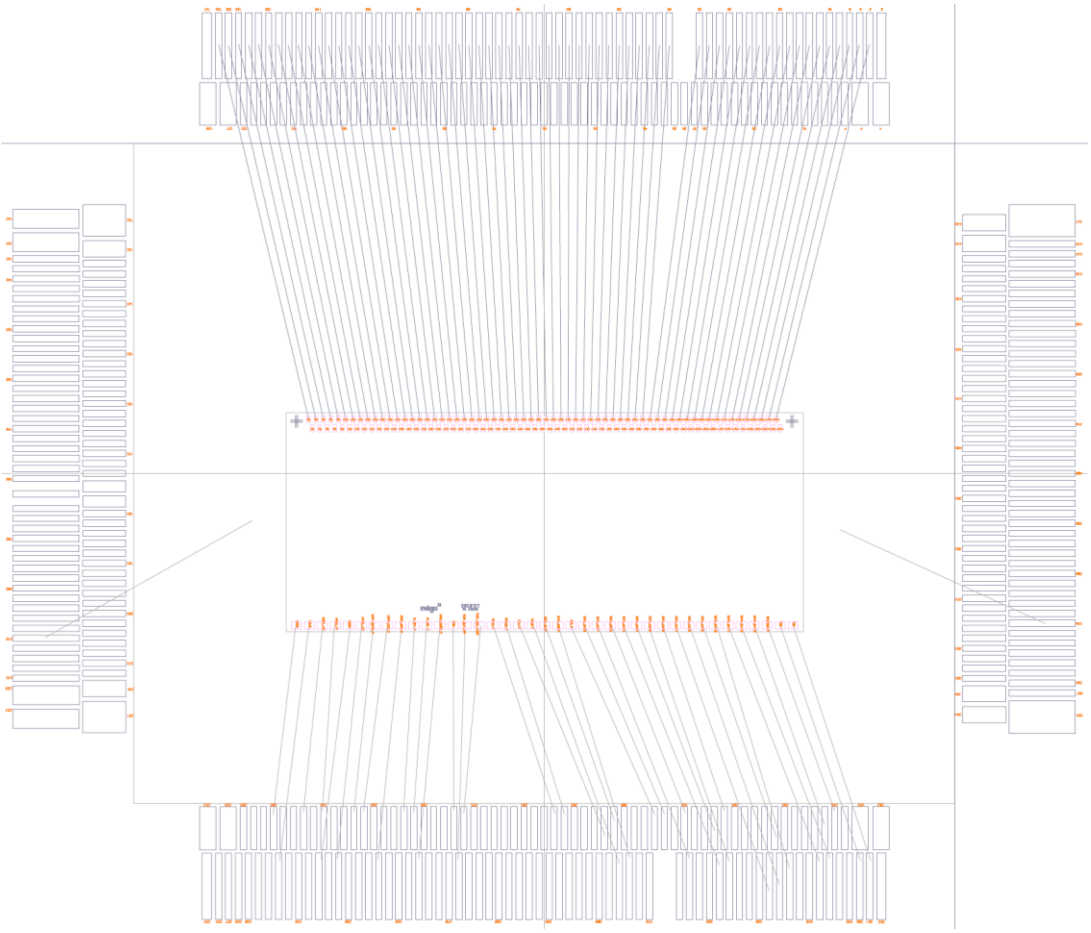


Figure 2.9 – Wire-bond diagram for Indigo with only 64 inputs connected

One important feature of this diagram is separating analog control inputs from digital outputs on the side with 39 pins. It will be very important when it comes to design a PCB layout as described in section 3.4.2.

By having the Indigo packaged in a PGA package, it is now time to move on to the next stage which is finding a socket for the package.

2.2.3 Socket for the PGA package

The Indigo chip is an expensive device. And it was time consuming and expensive to get the Silicon-die wire-bonded to the CPG44705 PGA package. So, it is wise to use a socket instead of soldering the Indigo package directly to a PCB.

Surprisingly our search for off-the-shelf socket led to nothing! In other words, even for a low quantity of less than ten, no supplier or manufacturer had them in stock. So, an order had to be placed. Three manufacturers were able to carry out the job, and considering the cost, Mill-max® was chosen to supply the part with a lead-time of four weeks!

2.2.4 The Indigo functionality

Before we delve into design issues, let us see how this Indigo chip works, to clarify for the reader why this ASIC has been chosen for this interface.

The Indigo chip has 128 independent inputs that can directly be connected to an imager array. Figure 2.10 shows block diagram of the internal circuits for each input [8].

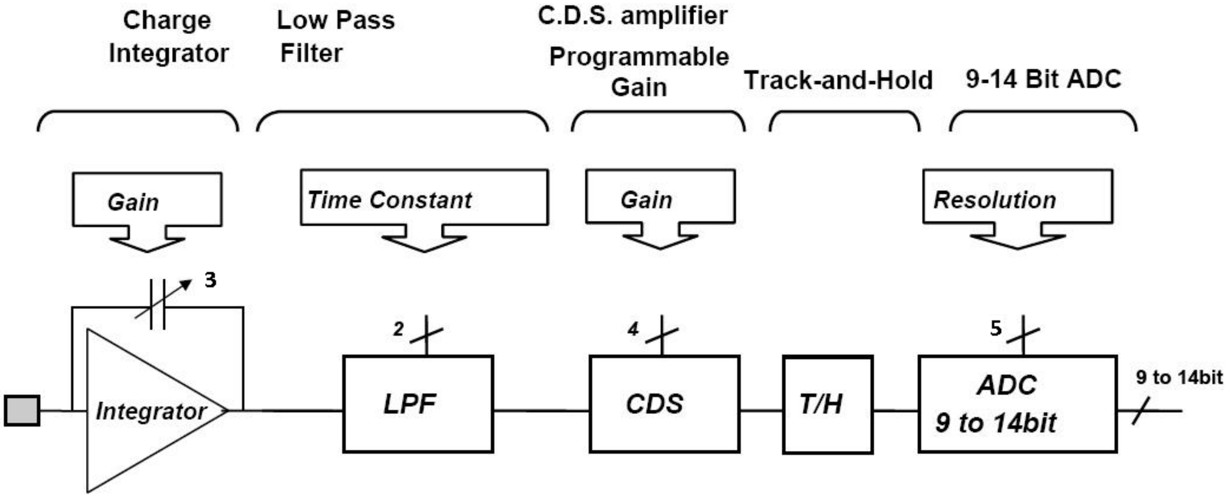


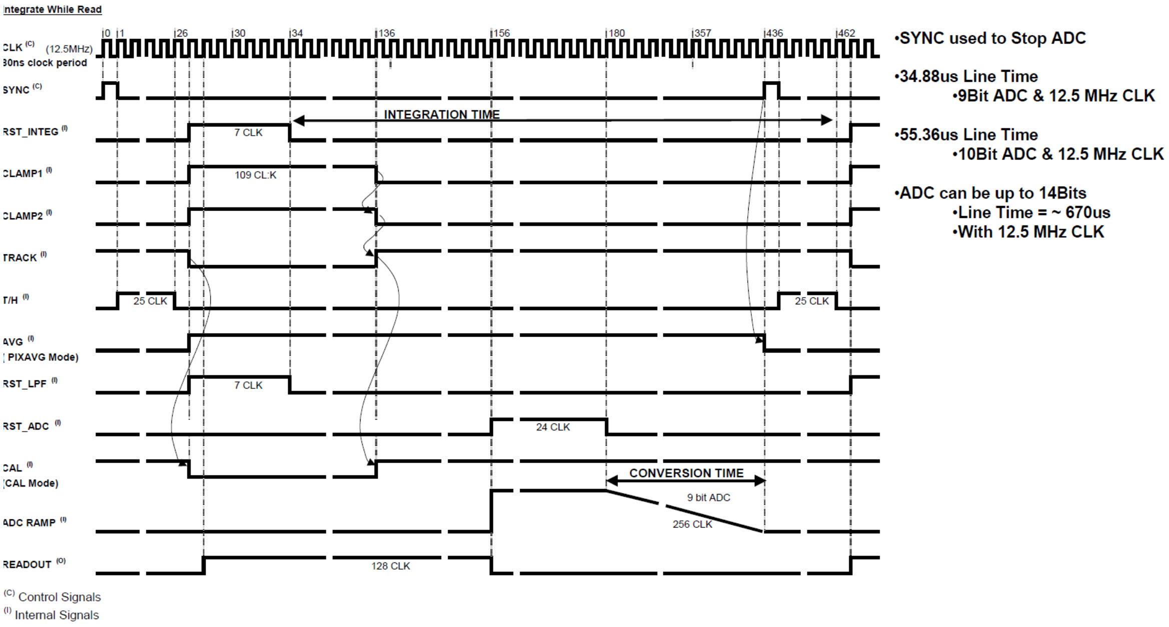
Figure 2.10 – Block diagram for each input pin of the Indigo chip [8]

The first block is an integrator with programmable gain. The second block is a low pass filter, again with programmable cut-off frequency. The next block is a correlated double-sampling module with programmable gain. Then there is a track and hold block. The last block is an analog to digital converter with programmable resolution from 9 to 14 bits. This output byte, that can be 9 to 14 bits width, is the output that should be sent to FPGA.

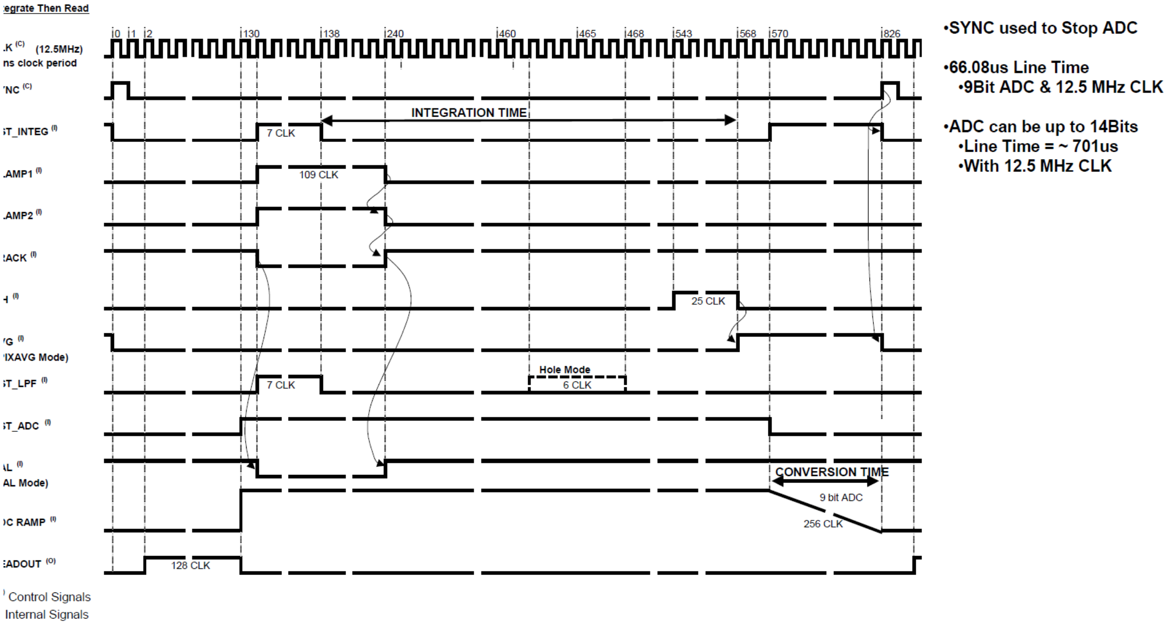
There are several modes of operation for this chip:

- (IWR) Integrate while read
- (ITR) Integrate then read
- (MAMO) Mammography

Figure 2.11(A) shows timings for one cycle in IWR mode [8], and Figure 2.11(B) shows the timings for ITR mode [8]. As inferred from names, the ITR mode has a longer period than the IWR mode, meaning that frame refreshing takes longer time in the ITR mode than IWR mode.



(A) IWR mode [8]



(B) ITR mode [8]

Figure 2.11 – Timing diagrams for the Indigo chip for IWR and ITR modes [8]

It can be seen in the figures 2.11(A) and 2.11(B) that in the IWR mode, integration and output reading are happening partially concurrently. But in ITR mode, integration time is quite separate from the output reading time.

2.3 Design of schematic diagram

Now that it is known how to use the Indigo chip, let us see how it can be embedded into the design of schematic diagram for the interface.

2.3.1 Opto-couplers for isolation

Digital input signals to gate-driver section come from FPGA. Also digital output signals coming out of the Indigo are sent back to FPGA for more processing. Figure 2.12 is similar to Figure 1.12 but with a marked analog section on the daughter-board PCB. The outputs of the imager array that are inputs to the Indigo are analog signals. In other words, charges travelling from the array outputs to the Indigo inputs, can have any value within a certain range. So, this part of the circuit should be taken in account as analog circuit. Therefore, the signals going to the array should be isolated from the digital noise and digital signals that are coming from FPGA.

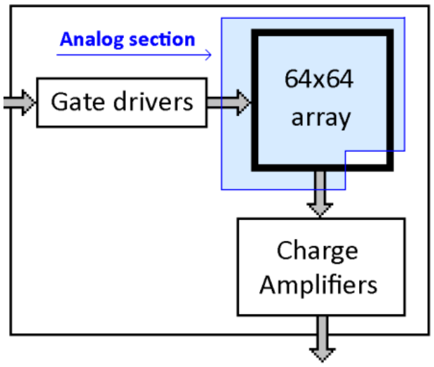


Figure 2.12 – Simplified block diagram of the interface

Given the above reasoning, it is inevitable to use opto-couplers for isolation. HCPL-314J offers two opto-couplers in a package. It also gives the possibility to change levels as well.

2.3.2 Multiplexers

As mentioned previously, in addition to charge amplifier section, another block of gate-drivers should also be designed. The gate driver section is mainly a de-multiplexer that chooses/activates one line/row at a time. Since our goal is a 64x64 array size, and because there is two signal lines going to each row of the detectors, then a 128x1 multiplexer should be designed. The IC AD1406 from Analog Devices provides 16x1 multiplexer in a small package such as TSSOP. Therefore eight AD1406 would provide us with the multiplexer size required.

2.3.3 Buffers

All digital input signals to the interface are coming from FPGA. Similarly all digital output signals of the interface are also going to the FPGA. To eliminate any loading effect on weak outputs coming from Cyclone II FPGA, a buffer stage is inserted as the first stage on the PCB. Also, in order to make sure that output signals going to FPGA have proper signal levels, a buffer stage is inserted at the output of the PCB. TC74VHC541FT from Toshiba provides eight fast (tpd = 3.7 ns) buffers in a small package like TSSO with supply voltage from 2.0V to 5.5V.

2.3.4 Digital to analog converters

The Indigo chip has seven analog voltage inputs that are used to control the operation of the chip [9]. The chip's manufacturer recommends providing them using digital to analog converters (DAC) with at least 12 bit resolution. Since these voltages will not be changed much, fast operation DACs are not required. In the electronic components market one can find very compact DACs with multiple units based on I^2C addressing method. But I^2C protocol is not the best way for us as FPGA is used for controlling these DACs. So, parallel addressing DACs should be searched for.

AD5725 from Analog Devices contains four DACs with 12 bit resolution voltage output and parallel addressing method. Figure 2.13 shows functional block diagram of the chip.

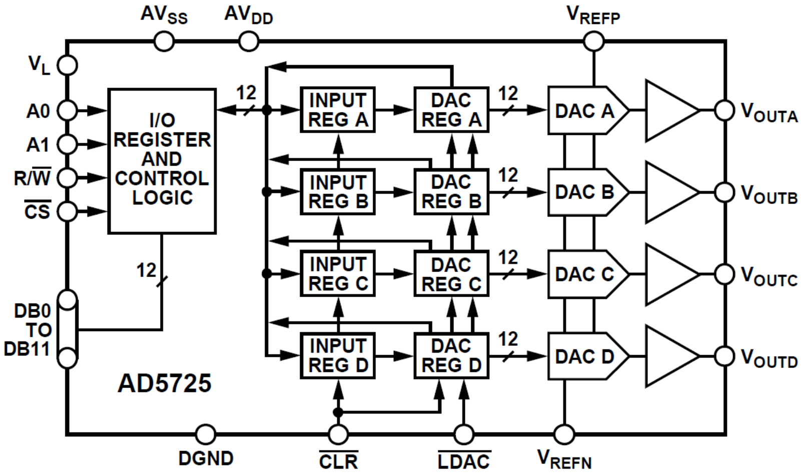


Figure 2.13 – Functional block diagram for AD5725 [9]

2.3.5 TTL to LVDS converter

The Indigo chip has three inputs of CLK, SYNC, and DATA_IN that are of LVDS (Low Voltage Differential Signal) type. This means for each of them, there are two lines of ‘signal’ and ‘signal_bar’ going into the Indigo. The ‘signal_bar’ is just inverted version of ‘signal’ line, for example CLK and CLKB (read CLK bar).

Since clock signal (CLK) is the fastest signal on the interface, then the TTL to LVDS converter must be a very fast operating chip. FIN1531 manufactured by Fairchild Semiconductor meets our requirements. It has four TTL to LVDS converter on chip.

2.3.6 Output driver stage for the Indigo chip

The Indigo outputs (14 lines) are of current type with 0.5 mA ±10% range. In order to transfer this output signal to the FPGA, we should first convert the output signal from current to voltage. Details of this section cannot be publicly disclosed based on the terms of an NDA we signed with the Indigo manufacturer to get access to this technical information.

After this stage, the outputs would be digital voltage type. In other words, a digital ‘0’ & ‘1’ on the output pin of the Indigo chip would be represented by 0V and 5V signals at the output of this stage, which in turn should be transferred to FPGA.

2.3.7 Daughter-board

Since this interface will be used by different STAR members to test different types of arrays, it should allow users to change the imager array. And these imager arrays are built in the G2N lab as prototype sample on Silicon or glass substrates. In most cases, these sample arrays are wire-bonded to packages or other external devices before they can be used. In this case, where the imager array can be as large as 64x64 pixels, dimensions of the imager array, depending on the pixel size, can be up to a few centimeters by a few centimeters, and if we add the area required for bond-pads, it would be even larger. There are packages on the market that can house such large size Silicon or glass dies, but they are very costly. So, it is in our benefit if we just wire-bond the imager array to a daughter-board that is mounted on the main PCB. In this way, different users will use different daughter-boards while the main PCB is the same for all of them. Figure 2.14 shows the idea of daughter-board mounted on main-board.

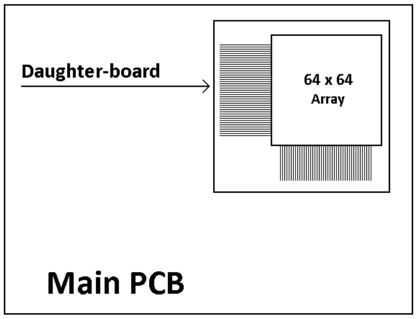


Figure 2.14 – Daughter-board mounted on main-board

2.3.8 Connector for daughter-board

Let us review how many lines need to be connected using this connector(s). For a 64x64 imager array, there are up to 128 lines on the gate-drive side (see Figure 1.12 –simplified block diagram) and 64 lines on the charge amplifier side. Since lines at the gate-driver side carry digital signal and lines at the charge amplifier side carry analog signals, it is better to separate them and use two connectors in order to reduce overall noise in the system. In this case it is obvious that the two connectors should have the same height.

Size of the connectors should also be compact, otherwise the size of the daughter-board will need to be increased unnecessarily. The size of the daughter-board that holds the imager array is two inches by two inches maximum (5 cm by 5 cm in metric system).

A search for the connectors led us to a few manufacturers for this part. We eventually chose Hirose connectors. Two connectors with 140 pins and 80 pins would meet all of our requirements. They both have a height of 5 mm, and the 140 pin connector has a length of 49.1 mm.

Inserting this 140 pin connector on the output lines of the imager array adds some parasitic capacitance associated with connector conductors to the lines. It would be a fixed pattern noise because the length and path of lines on daughter-board and main-board are fixed.

One solution to minimize this unwanted parasitic capacitor is to move the Indigo chip to the daughter-board. The drawback of this method is the need to have one Indigo socket on each daughter-board. After consulting with STAR group director, we decided to put more weight on the parasitic capacitance than the cost of one PGA socket per daughter-board. This is why we decided to put the Indigo chip on the daughter-board instead of the main-board. Figure 2.15 shows the design when the daughter-board hosts both imager array and Indigo chip.

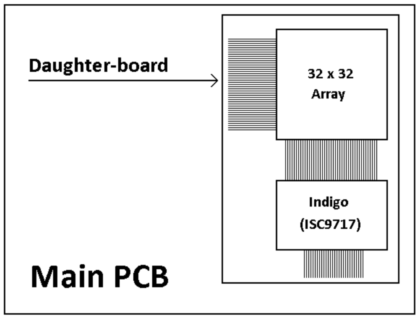


Figure 2.15 – Daughter-board hosting imager array and Indigo chip

The output of the daughter-board is digital signal. So, the parasitic capacitance associated with 80 pin connector is negligible.

2.3.9 Power supply tree

In order to design an efficient power supply tree, several factors should be considered, including maximum positive voltage required, minimum negative voltage required, and finally other voltages required. Maximum current needed for each voltage level has a critical importance in this design.

Here is a list of voltages required for this design:

- Variable +15 V or even +20V for gate-driver stage – Analog section
- Variable -15 V or even -20V for gate-driver stage – Analog section
- Fixed +5 V for the Indigo chip – Analog section
- Fixed +5 V for the Indigo chip and other TTL components – Digital section
- Fixed +3.3 V for buffers sending data back to FPGA – Digital section

In order to calculate current consumption for each voltage level, we can simply add nominal currents for all chips together. At the same time we should also consider maximum current levels for the chips as well.

Resistors and other electronic loads should also be considered when calculating maximum current levels. In this design there is no such high-current load and it would be safe to ignore them.

A safe margin should be considered above maximum nominal currents calculated in this way. Especially for high-current voltage levels when heat-sink is necessary for regulators, careful attention should be given to environmental assumptions for the regulator IC and its heat-sink.

Another feature that would greatly enhance the interface portability is a single-pole power supply. This means any negative voltage should be created from positive levels. This is not difficult to meet given a wide range of DC-to-DC converter ICs available.

Table 2.1 shows sum of maximum currents for different voltage levels based on information extracted from component's datasheets.

Table 2.1 – Current levels required for voltage sources

Voltage level	Maximum current level
Variable +15 V	200 mA
Variable -15 V	20 mA
+5 V Analog	100 mA
+5 V Digital	1000 mA
+3.3 V Digital	50 mA

It is not surprising that the highest current source is required for digital +5V as most of the components on the board are supplied by this source. And the minimum level is required by -20 V variable source. This voltage level is required only to apply negative voltage to gate-driver signals when they are off. This negative voltage at off state helps TFTs to prevent from V_T shifting.

Figure 2.16 shows a high-level look at the power supply blocks and how they are fed through different stages.

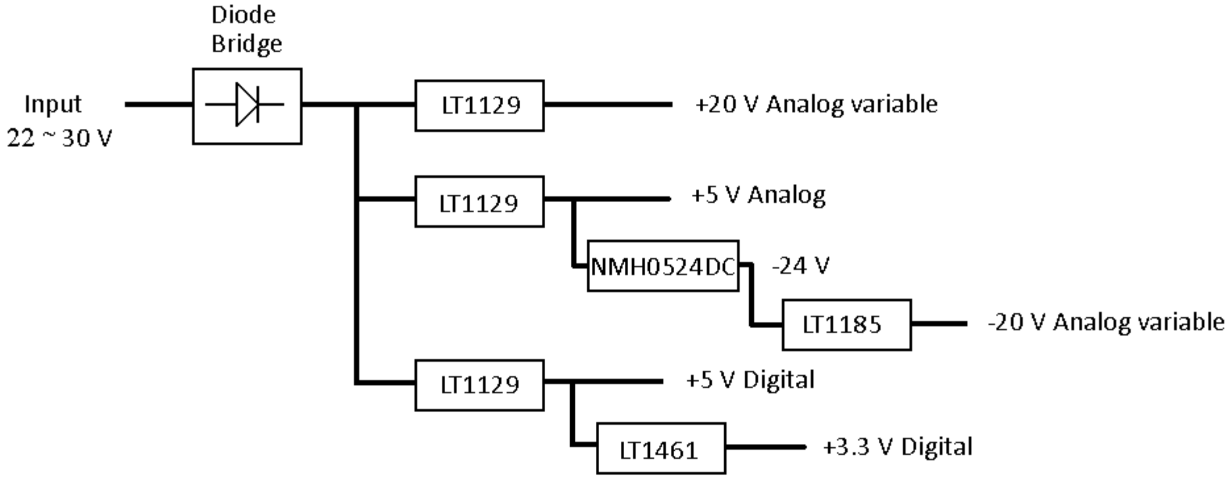


Figure 2.16 – Power supply tree

This figure shows minimum input voltage of 22 volts. In reality it can be lower than that. In fact the main input voltage should be at least two volts higher than the maximum required analog variable voltage. For example, if maximum required analog voltage is +15V, then the interface can be supplied by a voltage between +17 V to +30 V.

There is one diode bridge as the first block right after the input terminals. It is there to protect the rest of the circuit against accidental connection to wrong polarity supply voltage.

Then there are three LDO voltage regulators LT1129. This voltage regulator made by Linear Technology® can provide a positive output voltage from 3.8V to 30V using a few external components. Figure 2.17 shows a typical schematic diagram for this chip [10].

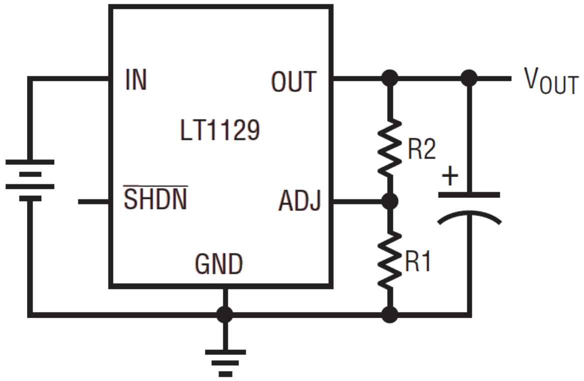


Figure 2.17 – Typical circuit for LT1129 with external components [10]

The LT1129 chip, like many other voltage regulators comes in different packages. The differences between these packages are size, shape, power rating (equivalent to current rating), and heat dissipation means. Considering the margins given in Table 2.1 it is clear that the best type for +5 V digital is Q package. For the other two LT1129, one can use different package type. But since space is not that crucial for us from one side, and to preserve uniformity of the design from other side, it is suggested to use the same Q package even for the other two LT1129 regulators. In this way, diversity of required components would be minimal.

The other two Linear Technology voltage regulators of LT1185 and LT1461 have similar structure. In other words, they can be embedded by a few external resistors and capacitors to construct a very simple voltage regulator. LT1185 is a negative voltage regulator.

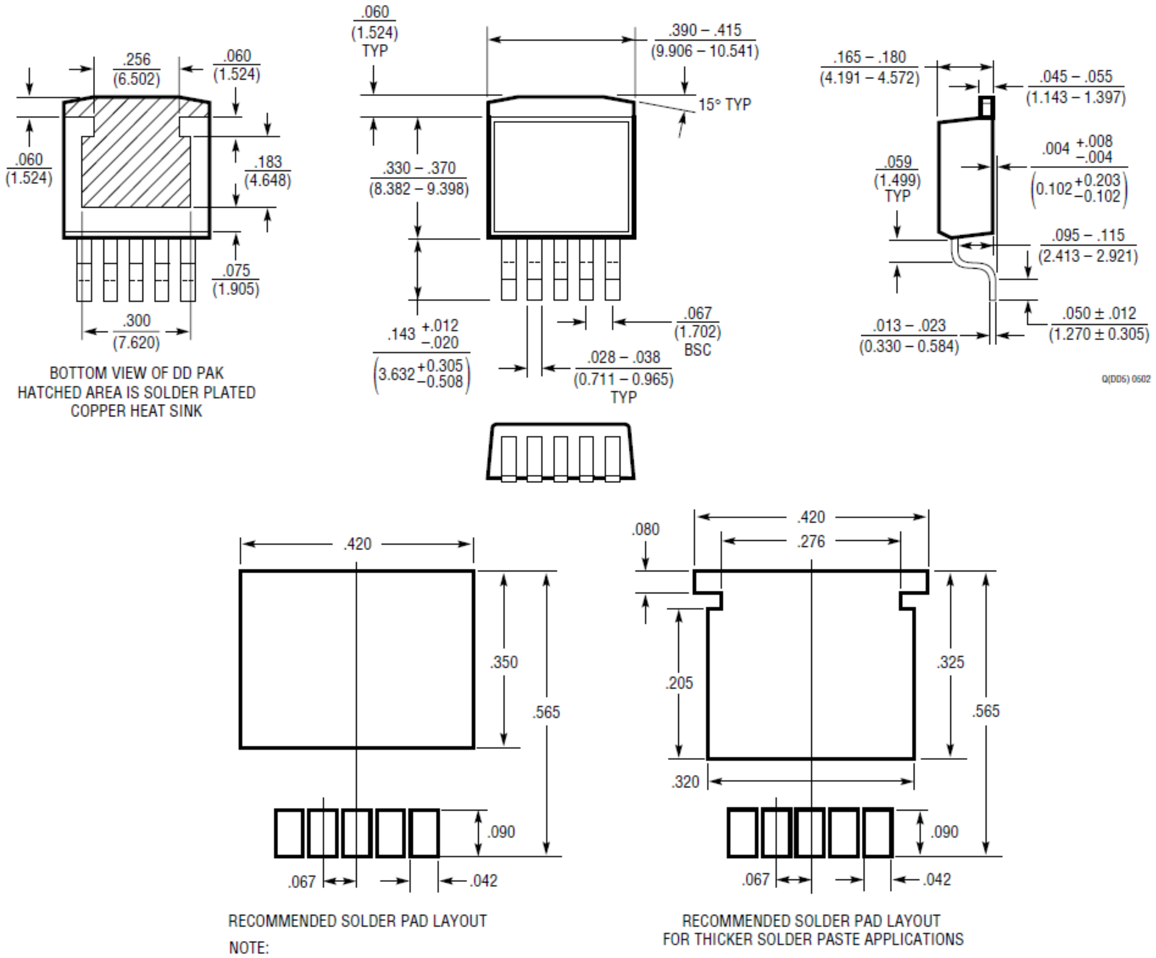
All these Linear Technology regulators are of switching type. Therefore the values of external capacitors have a direct effect on the frequency of oscillation. Care should be taken in choosing these capacitor values. More detailed information can be found in [10], [11], and [12].

Finally, the last component to notice here is the NMH0524DC chip. It is an Isolated 2W Dual Output DC/DC Converter. The input to this IC is a single pole +5V and the IC provides a fixed dual output at ±24 V which is isolated from the input. It is only the -24 V that is used in this interface.

The output current level of the chip is relatively low at ±42 mA. But since the current required for negative voltage level is low (around 20 mA), it meets the specifications.

An LT1185 then is used to change this fixed -24 V level to a variable -20 V level using a potentiometer.

Finally, for consistency purpose as well as current providing capability, SMD package Q was chosen for all regulator ICs except the NMH0524DC which is available only in DIP format. Figure 2.18 shows the chosen package for Linear ® regulators [10].



RECOMMENDED SOLDER PAD LAYOUT

RECOMMENDED SOLDER PAD LAYOUT
FOR THICKER SOLDER PASTE APPLICATIONS

NOTE:
1. DIMENSIONS IN INCH/(MILLIMETER)
2. DRAWING NOT TO SCALE

Figure 2.18 – Package chosen for LDO regulators [10]

2.4 Design of PCB layout

Once design of the schematic diagram is completed, PCB layout can be started. There are two PCBs to design, the main PCB and the daughter-board.

2.4.1 Main PCB

The main PCB hosts gate-drivers, DACs, buffers, LVDS-to-TTL adapters, daughter-board, and other components that comprise both analog and digital signals. In addition to these sections, it consists of a power supply section that creates all voltage levels required for operation of the interface.

2.4.1.1 Separation of analog and digital sections

In digital systems there are fast rising and falling edges often followed by some small domain ringing. A digital system, due to its nature, recognizes voltages above NMHL (noise margin for high input – lower level) as ‘1’ state, and voltages below NMLH (noise margin for low input – higher level) as ‘0’ state. Therefore it is immune to unwanted signals that are added to waveforms as long as they don’t change the state of a signal.

Contrary to digital systems, in an analog system, such fast rising & falling edges or unwanted signals added to waveforms contribute to increasing noise level at output. This is why when it comes to design a mixed-signal PCB layout, care must be given to separating the analog section from the digital section. Figure 2.19 shows a view of the border between analog section and digital section on the path from FPGA to gate-drivers to Indigo chip. Opto-couplers are located right on the border.

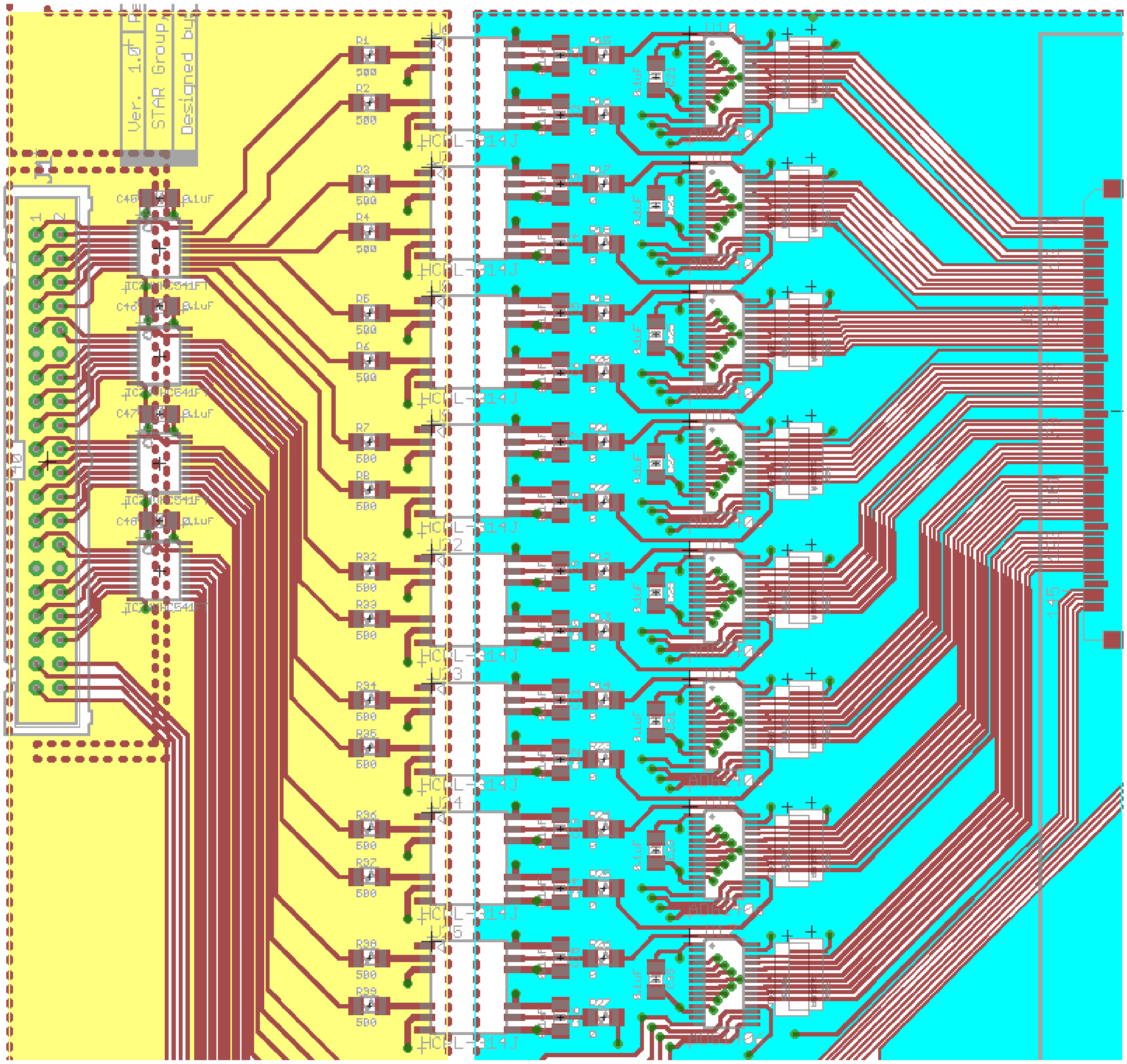


Figure 2.19 – Separation of analog (blue) and digital (yellow) sections by opto-couplers

The Indigo outputs are digital signals. Therefore they do not need to be isolated.

2.4.2 Daughter-board

The daughter-board PCB hosts the Indigo chip which has both analog and digital sections. All 64 inputs to the Indigo are considered as analog signal. On the other side with 39 pins, there are both digital and analog sections. So, care must be taken to dividing the area under the Indigo footprint to analog and digital sections. Figure 2.20 shows a top view of the layout.

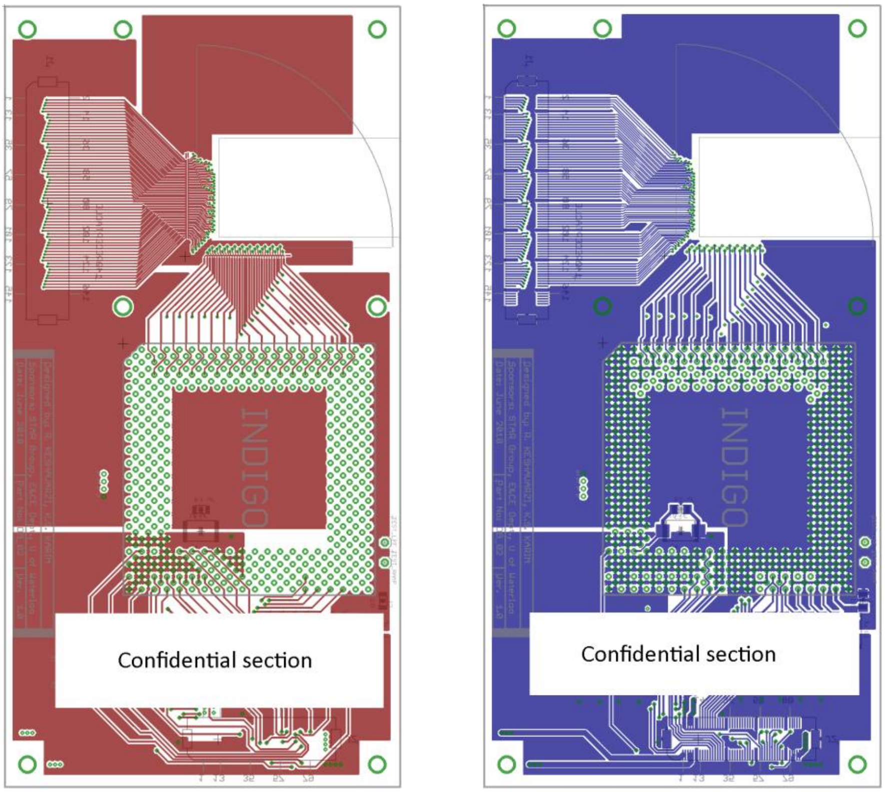


Figure 2.20 – Top and bottom view layout for the daughter-board

2.4.3 Daughter-board for Indigo characterization

Since there was no previous experience with the Indigo chip utilization in STAR group, and before the Indigo chip is used to readout an X-ray detector array, it was necessary to make sure that thorough knowledge of programming and utilizing of Indigo chip is in hand. Therefore a custom board designed specifically for Indigo characterization. Figure 2.21 shows the layout for this PCB.

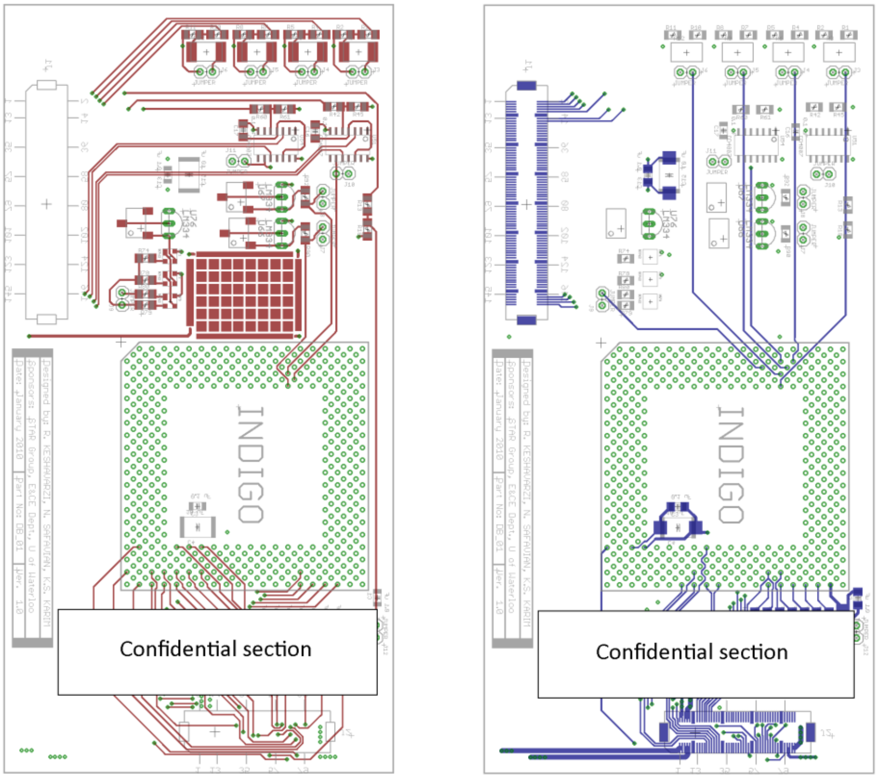


Figure 2.21 – Layout for daughter-board used for Indigo characterization

2.4.4 Multi-layer PCB

A PCB is a means to connect different point in a circuit together. When the number of connections to make goes up, a single or double layer PCB will not be able to provide all the connection. In such cases using a multi-layer PCB is inevitable.

In this design, both main-board and daughter-board use four layers PCB. One inner layer is dedicated to the supply routes. Another inner layer is dedicated to ground plane. The two outer layers are dedicated to signal routes in order to provide access to signals for probing in case of necessity for troubleshooting.

To avoid ground loops in the circuit, the digital ground and analog ground should meet only in one spot. Having a ground loop causes some current in the loop which will increase noise level in the system.

Chapter 3 MATLAB and VHDL coding

Once the main PCB and its daughter-board are manufactured and populated with electronic components, it should be connected to Altera DE2 board which contains one Cyclone II FPGA. The FPGA provides a flexible configurable hardware platform that interacts with the interface from one side, and a personal computer from another side. VHDL language is used for FPGA programming under Quartus® software environment.

On the computer side, a program written in MATLAB® language receives the information sent by the FPGA over a serial cable using RS-232 protocol byte by byte and puts them together to create the image captured by the Indigo chip.

3.1 VHDL scripting

The FPGA on the DE2 board has several functions:

- Programs the on-board DACs for Indigo operation
- Programs the Indigo chip itself
- Interacts with user through switches, push-buttons, and seven-segment displays to control different aspects of the Indigo chip
- Creates necessary pulses for the gate-driver block (Figure 2.11)
- Receives the high-speed data stream sent by main-PCB
- Stores data in FIFO memory for one frame
- Retrieves data from FIFO memory and sends them to PC over serial cable at a slower rate

Details of each task done by FPGA are provided below.

3.1.1 Digital-to-analog converter's programming

Each AD5725 chip contains four independent 12 bit parallel input DACs with voltage outputs[9]. Figure 3.1 shows pulses required for a programming cycle.

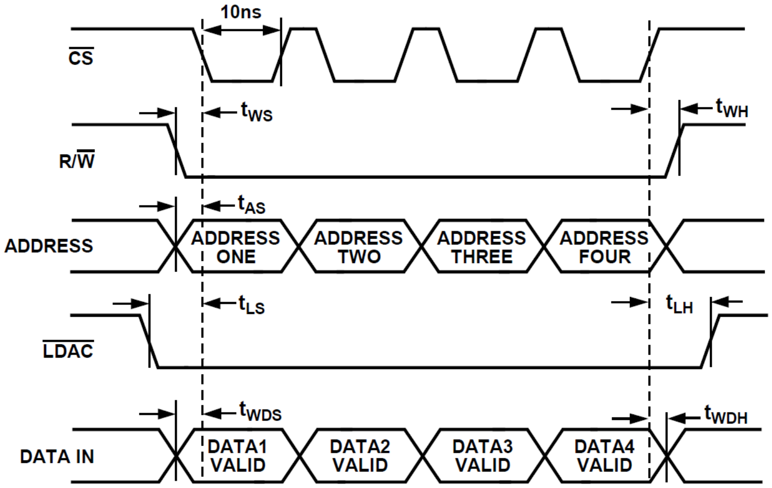
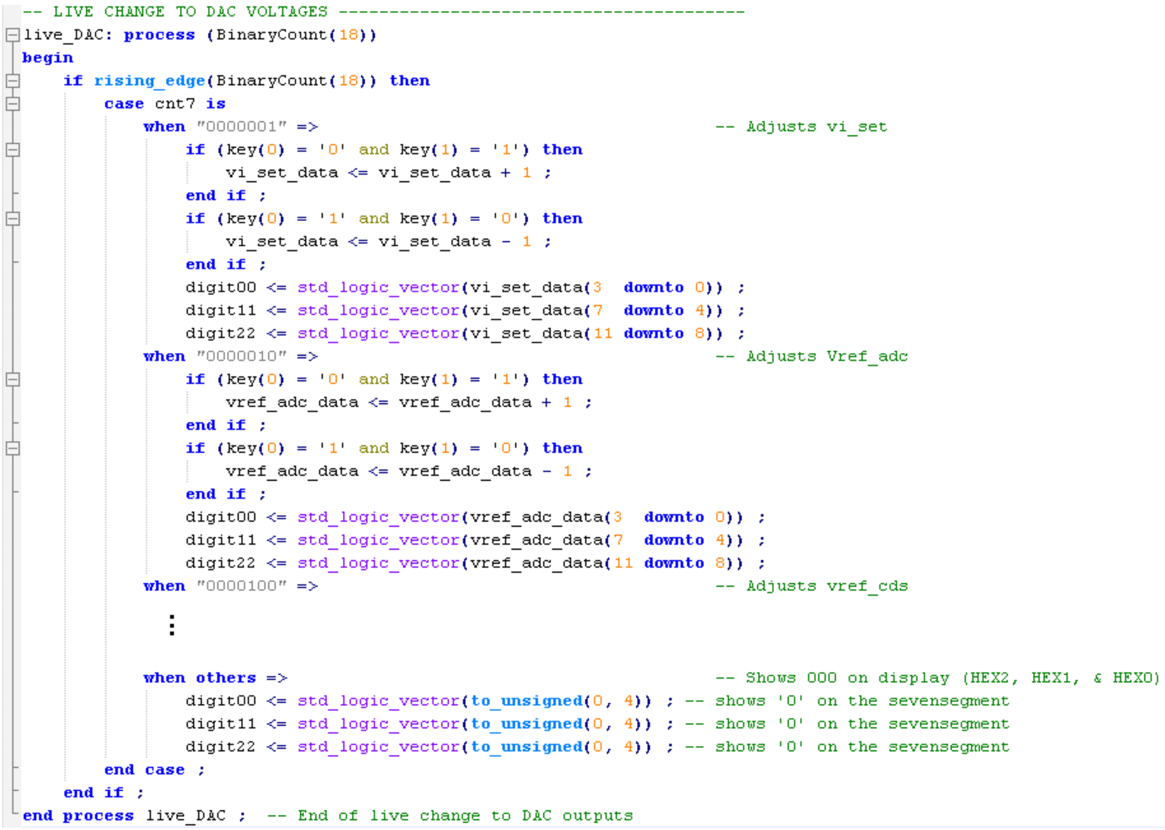


Figure 3.1 – Programming cycle [9]

These pulses are generated by the FPGA. For programming purpose, eight GPIO lines must be dedicated to the two AD5725, four to each one.

For normal operation of the Indigo chip, one time programming of the DACs is enough. But during the development and tuning of the prototype system, it was necessary to vary the analog voltages applied to the chip.

In order to have a close look at a sample part of the VHDL code, Figure 3.2 shows a code segment of a simple mechanism that provides real time control over all seven analog voltages applied to the Indigo chip. The push button key[0] increments the DAC value and key[1] decrements the associated DAC value. Only one voltage is varied at a time which is selected through switches sw[0] to sw[6].



```
-- LIVE CHANGE TO DAC VOLTAGES ----------------------------------------
live_DAC: process (BinaryCount(18))
begin
    if rising_edge(BinaryCount(18)) then
        case cnt7 is
            when "0000001" =>                                   -- Adjusts vi_set
                if (key(0) = '0' and key(1) = '1') then
                    vi_set_data <= vi_set_data + 1 ;
                end if ;
                if (key(0) = '1' and key(1) = '0') then
                    vi_set_data <= vi_set_data - 1 ;
                end if ;
                digit00 <= std_logic_vector(vi_set_data(3  downto 0)) ;
                digit11 <= std_logic_vector(vi_set_data(7  downto 4)) ;
                digit22 <= std_logic_vector(vi_set_data(11 downto 8)) ;
            when "0000010" =>                                   -- Adjusts Vref_adc
                if (key(0) = '0' and key(1) = '1') then
                    vref_adc_data <= vref_adc_data + 1 ;
                end if ;
                if (key(0) = '1' and key(1) = '0') then
                    vref_adc_data <= vref_adc_data - 1 ;
                end if ;
                digit00 <= std_logic_vector(vref_adc_data(3  downto 0)) ;
                digit11 <= std_logic_vector(vref_adc_data(7  downto 4)) ;
                digit22 <= std_logic_vector(vref_adc_data(11 downto 8)) ;
            when "0000100" =>                                   -- Adjusts vref_cds

              ⋮

            when others =>                                      -- Shows 000 on display (HEX2, HEX1, & HEX0)
                digit00 <= std_logic_vector(to_unsigned(0, 4)) ; -- shows '0' on the sevensegment
                digit11 <= std_logic_vector(to_unsigned(0, 4)) ; -- shows '0' on the sevensegment
                digit22 <= std_logic_vector(to_unsigned(0, 4)) ; -- shows '0' on the sevensegment
        end case ;
    end if ;
end process live_DAC ;  -- End of live change to DAC outputs
```

Figure 3.2 – Real-time change of analog voltages applied to the Indigo chip

The switch roles and respective voltages are:

Sw[0] for Vi_set

Sw[1] for Vref_adc

Sw[2] for Vref_cds

Sw[3] for Vref_integ

Sw[4] for Vi_lpf

Sw[5] for Vadj_ramp

Sw[6] for Vcal

In practice, the operation of this piece of code was very effective and essential to operate the Indigo chip. It enables the user of the interface to tune the Indigo chip to its best work point.

3.1.2 Indigo programming

The Indigo chip is serially programmed using the DATA_IN input. Gains of the different blocks shown in Figure 2.10 are determined in programming cycle. One of the GPIO lines on the DE2 board is assigned to this signal.

Since the programming details are confidential due to NDA signed between STAR group and the Indigo manufacturer, no additional information is presented here.

3.1.3 Grey versus binary code

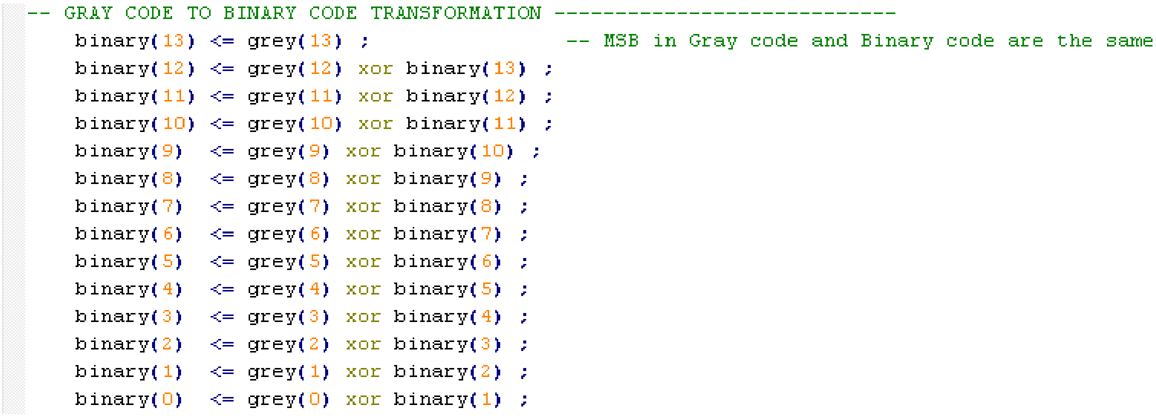
Output of the Indigo chip is grey coded. A grey code changes only one bit at a time when it is incremented continuously. For example, Table 3.1 shows grey codes and binary codes for a four bits number system.

Table 3.1 – Gray and Binary codes for four bit numbers

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

As it can be seen from Table 3.1, grey code has fewer transitions between ‘0’ and ‘1’ compared to binary code. This is a good feature for high-speed systems which avoids unnecessary switches.

Since most digital systems work with binary system, it is necessary to change the grey code to binary code. Fortunately it is not difficult to do so. Just bitwise exclusive-or (xor) logical operation would do the job. This transform can be done either by VHDL code on the FPGA or in MATLAB code on the PC side. But it is more efficient to do it using hardware (VHDL code) that has parallel structure. Figure 3.3 shows related VHDL code.



```
-- GRAY CODE TO BINARY CODE TRANSFORMATION ---------------------------
    binary(13) <= grey(13) ;                  -- MSB in Gray code and Binary code are the same
    binary(12) <= grey(12) xor binary(13) ;
    binary(11) <= grey(11) xor binary(12) ;
    binary(10) <= grey(10) xor binary(11) ;
    binary(9)  <= grey(9) xor binary(10) ;
    binary(8)  <= grey(8) xor binary(9) ;
    binary(7)  <= grey(7) xor binary(8) ;
    binary(6)  <= grey(6) xor binary(7) ;
    binary(5)  <= grey(5) xor binary(6) ;
    binary(4)  <= grey(4) xor binary(5) ;
    binary(3)  <= grey(3) xor binary(4) ;
    binary(2)  <= grey(2) xor binary(3) ;
    binary(1)  <= grey(1) xor binary(2) ;
    binary(0)  <= grey(0) xor binary(1) ;
```

Figure 3.3 – VHDL code for grey code to binary code transform

3.1.4 Gate drivers pulses

Gate lines on the array are excited through pulses created by the FPGA. Each gate line is connected to a GPIO line on the DE2 board. The gate driver pulses are generated in daisy-chain format. In other words, one row at a time is stimulated. Figure 3.4 shows sample pulses for gate-drivers.

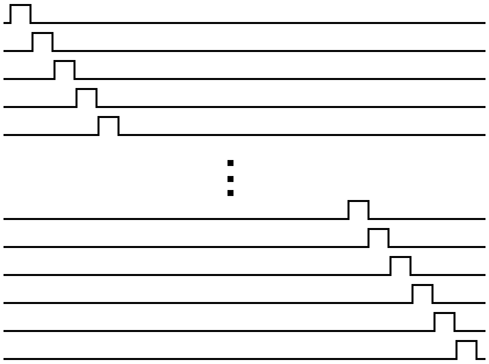


Figure 3.4 – Gate-driver pulses generated by FPGA

It is quite clear that these pulses must be synchronous with the Indigo operation. Each pulse here allows the pixels in one row of the array to discharge into the input capacitor of Indigo pins.

3.1.5 UART handling

In order to send data received from the Indigo chip, a data path must be established between the DE2 board and the personal computer. It would be much faster through USB port, however, it would be a difficult challenge and beyond the scope of this project. A serial port communication using RS-232 protocol would be much easier. Figure 3.5 shows this port on DE2 board [13].

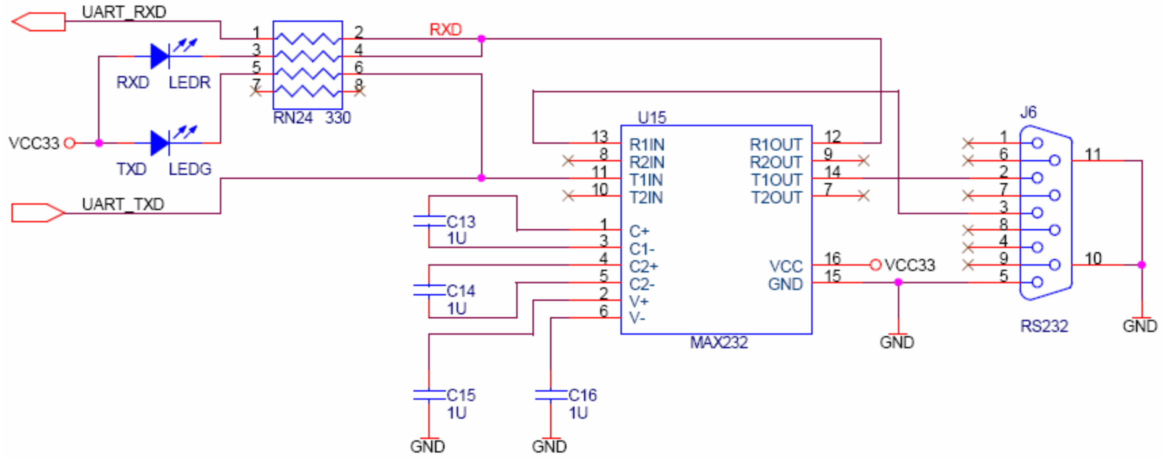


Figure 3.5 – Serial port interface on DE2 board [13]

As shown in this figure, there are two signals of UART_RXD and UART_TXD that are used for receiving and sending data bits.

To facilitate faster development of the interface, a piece of VHDL code written by a French company (www.alse-fr.com) was used and modified to match our needs. Written permission of the owner of the code has been acquired for the purpose of this research.

The original UART code has limited capabilities, for example only two speeds of 9600 bps or 11500 bps were supported, or the data byte width was only eight bits. The modified VHDL code supports several speed levels of 9600 bps, 19200 bps, 38400 bps, 57600 bps, and 115200 bps. For byte size, seven bits is a better option for this project because the Indigo output can be a maximum of fourteen bits. In other words, one Indigo output with fourteen bits can be sent to PC using UART by two bytes of seven bits wide.

3.2 MATLAB coding

Once the FPGA is programmed and the interface started functioning, data packets are received at the computer's serial port. This packet of data was sent from Indigo to the FGPA first, and then travelled from the FPGA over a serial cable to reach the computer's port. Then a program running on the PC receives these packets of data and puts them in order. This program must be fully synchronized with the FPGA functionality. A single bit out-of-synchronization data transmission is enough to distort the captured image drastically.

3.2.1 UART handling

The UART can be configured to do some parity check to increase reliability of the data communication. In addition to this feature, it would be necessary to implement a mechanism that guarantees error-free data communication line by line, and frame by frame for each array. To do this, a byte of null was added at the end of each stream of data for pixels located on one line of the array. In other words, at the end of data sent to the PC for one line of pixels, a null word (14 bit of '0') is sent to PC as an end of line mark. It is worth mentioning that the Indigo chip would never create a null output under normal mode of operation.

Similarly, two words of nulls are used to mark an end of frame. In other words, the next word received after the two nulls mark, would be for the first pixel of the first row.

Handling serial ports is relatively easy in MATLAB. An object is defined for this purpose. Once the port is opened, data are stored in a buffer vector with variable length. This buffer can be read by the main program to get access to data received over the serial cable.

3.2.2 Graphical representation

Once the buffer data is read by the main program, it must be checked against null words and double null word. If they are not in the predicted place on this buffer, then there has been some data lost in the communication. Otherwise the null words should just be ignored. Then the rest of data in the buffer are actually data values read by the Indigo chip.

The next step would be storing this data vector into a matrix with the size of the array. From this point on, each number in the matrix represents gray-level at each pixel.

Finally the information in this matrix must be graphically presented on the monitor. The command `imshow(I,[low high]),` in MATLAB displays the grayscale image `I`, specifying the display range for `I` in `[low high]`. The value low (and any value less than low) displays as black; the value high (and any value greater than high) displays as white. Values in between are displayed as intermediate shades of gray, using the default number of gray levels [14].

Chapter 4 Test results

In order to test the readout circuitry as a whole, all system components must be put together as shown in Figure 4.1. The daughter-board is mounted on the main PCB. An Altera DE2® board connects the main PCB to a personal computer.

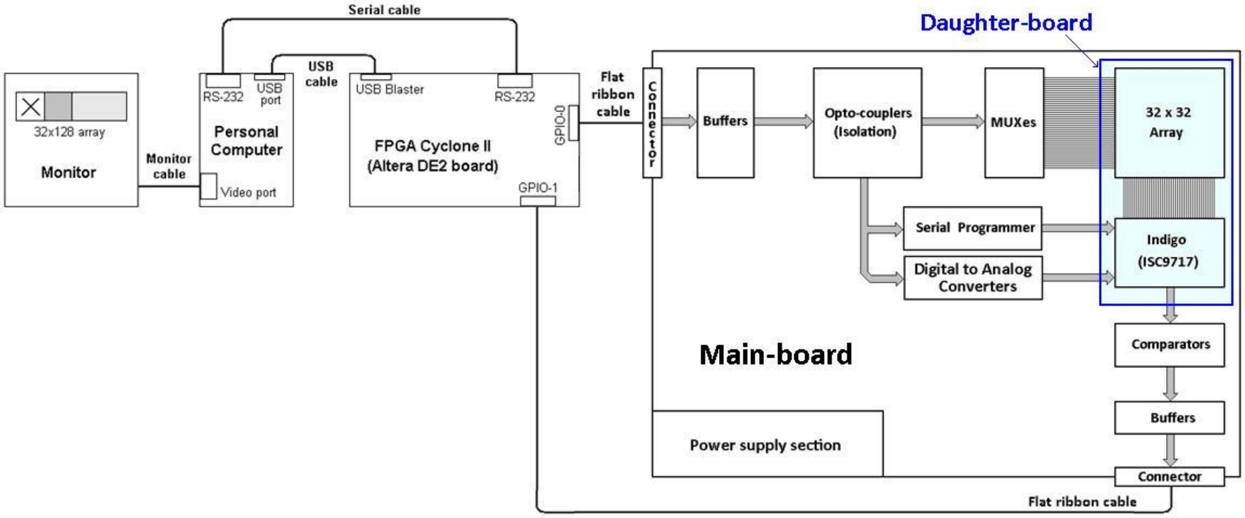


Figure 4.1 – Block diagram for the whole system

This is how the system works:

- The FPGA is programmed using the personal computer. It programs the Indigo chip as well as the on-board DACs, generates pulses applied to the gate driver, receives the fast-stream of data coming from the main PCB, and transfers the data to the PC over a serial cable (RS-232). The slow speed of data transfer over serial cable is the main limiting factor in the speed of frame capture by the system.

- A MATLAB program running on the personal computer receives the data sent from the FPGA.

4.1 Data transfer from Indigo to FPGA, and from FPGA to PC

Before we trust the pattern generated by this system on the computer's monitor as an X-ray frame captured, we must make sure that all data bits are sent and received in a matching order. Just a single bit mismatch is enough to distort the generated pattern.

Once way to acquire this trust is by sending known information to the FPGA and the personal computer, and expecting to see a matching known pattern for that known data-stream. For example, for the current 32x32 array in hand, instead of sending data generated by Indigo chip to FPGA, a column number can be sent for each pixel. In other words, for all 32 pixels in column 1, a data 01 is sent to FPGA. Similarly for all 32 bits in column 2, a data 02 is sent to FPGA. And this pattern would continue up to column 32 where a data 32 is sent to FPGA for all 32 pixels. Since the data sent to FPGA would be considered as coming from Indigo and therefore representing gray-level for each pixel, then an incrementing pattern with respect to column numbers would generate an incrementing brightness (gray-level) from left to right. Figure 4.2 shows a captured pattern for this scenario.

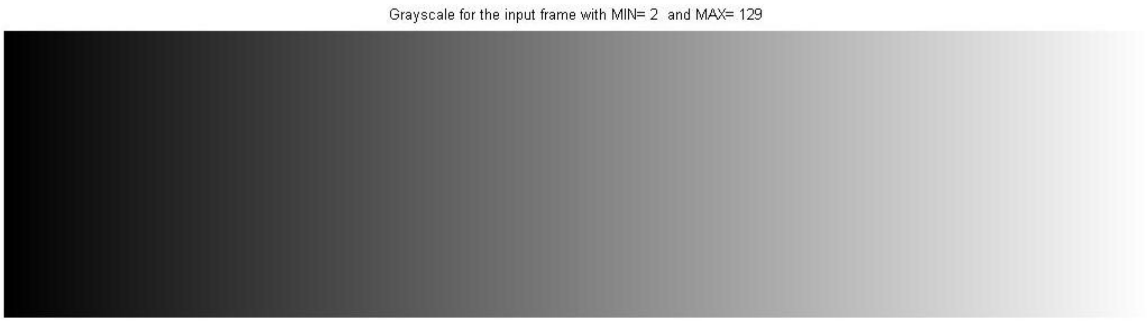


Figure 4.2 – Pattern generated proportional to incrementing column number

All 32 pixels in one column have the same column number. Therefore all of them have the same brightness (gray-level) in this figure.

The Indigo chip has 128 inputs. But as discussed in previous chapter (Section 2.2.2), only 64 of its inputs are wire-bonded to the package, and 64 of them are not. So, it would be nice if they are separated. Figure 4.3 shows the captured pattern for this case.

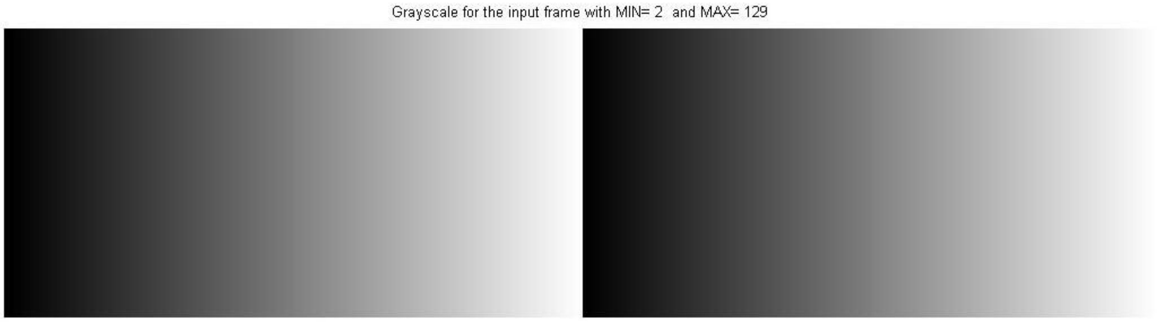


Figure 4.3 – Wire-bonded inputs are separated from not-connected ones

A similar pattern can be generated for row numbers. In such case all 32 pixels in a row would generate the same brightness (gray-level). Figure 4.4 shows the generated pattern for an incrementing row number.

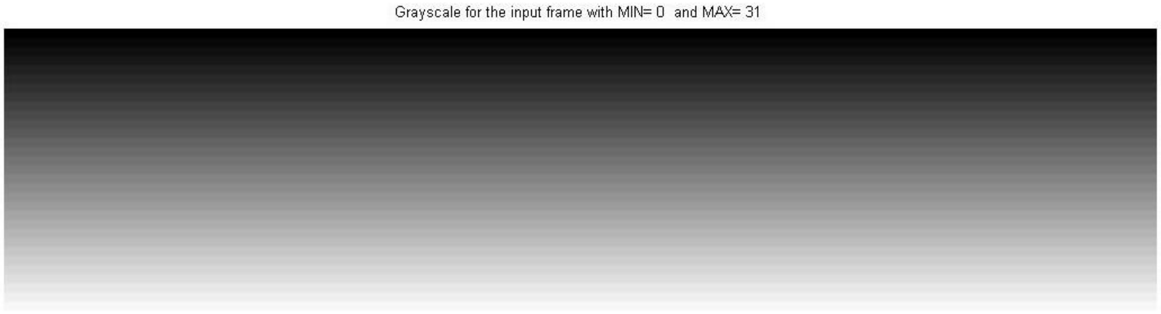


Figure 4.4 – Pattern generated proportional to incrementing row number

As expected, the brightness (gray-level) is incremented as row number increased.

Let us now combine the above two horizontal and vertical patterns. The data sent to FPGA is incremented by both row and column numbers. Figure 4.5 shows the data sent to FPGA and corresponding pattern generated.

Given the above patterns generated in accordance with expected results, it shows that all data is transferred between the main PCB and the FPGA and then between the FPGA and the personal computer in proper order with no lost or misplaced data bit. This shows that the VHDL and the MATLAB codes are functioning properly.

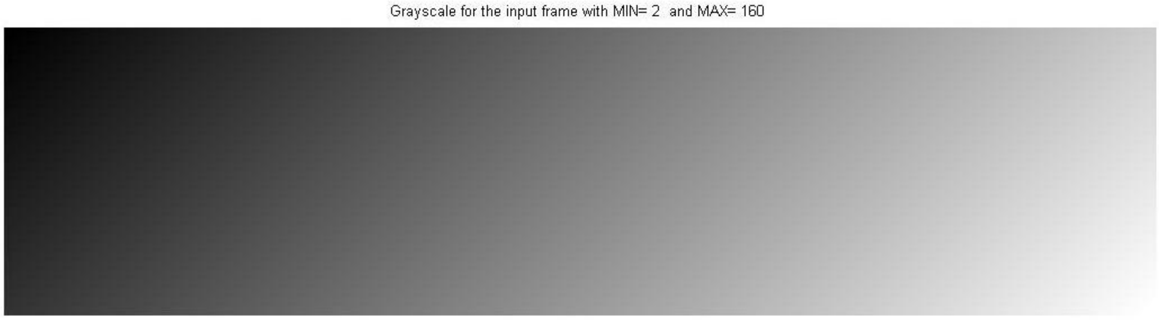


Figure 4.5 – Pattern generated proportional to incrementing row & column number

4.2 System components

Figure 4.6 shows the main PCB. The daughter-board is mounted on this board.

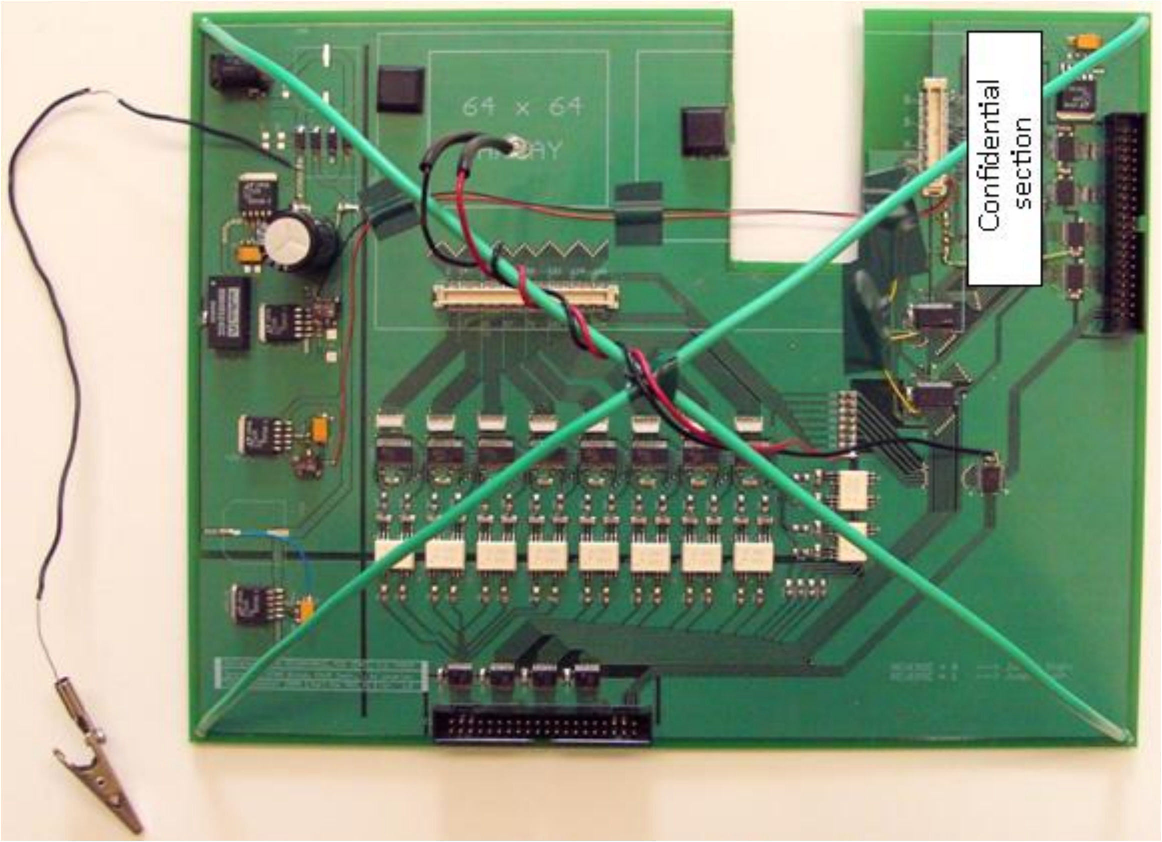


Figure 4.6 – Main PCB

Figure 4.7 shows the daughter-board that hosts an Indigo chip and a 32x32 PPS array.

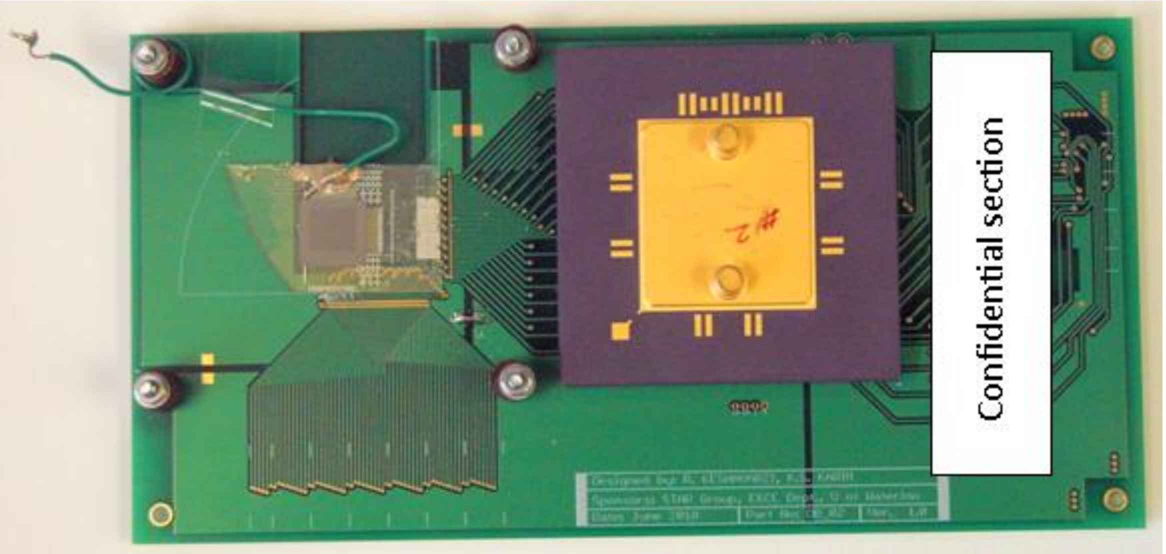


Figure 4.7 – Daughter-board PCB for 32x32 PPS array

Figure 4.8 shows another PCB designed solely for characterization of the Indigo chip.

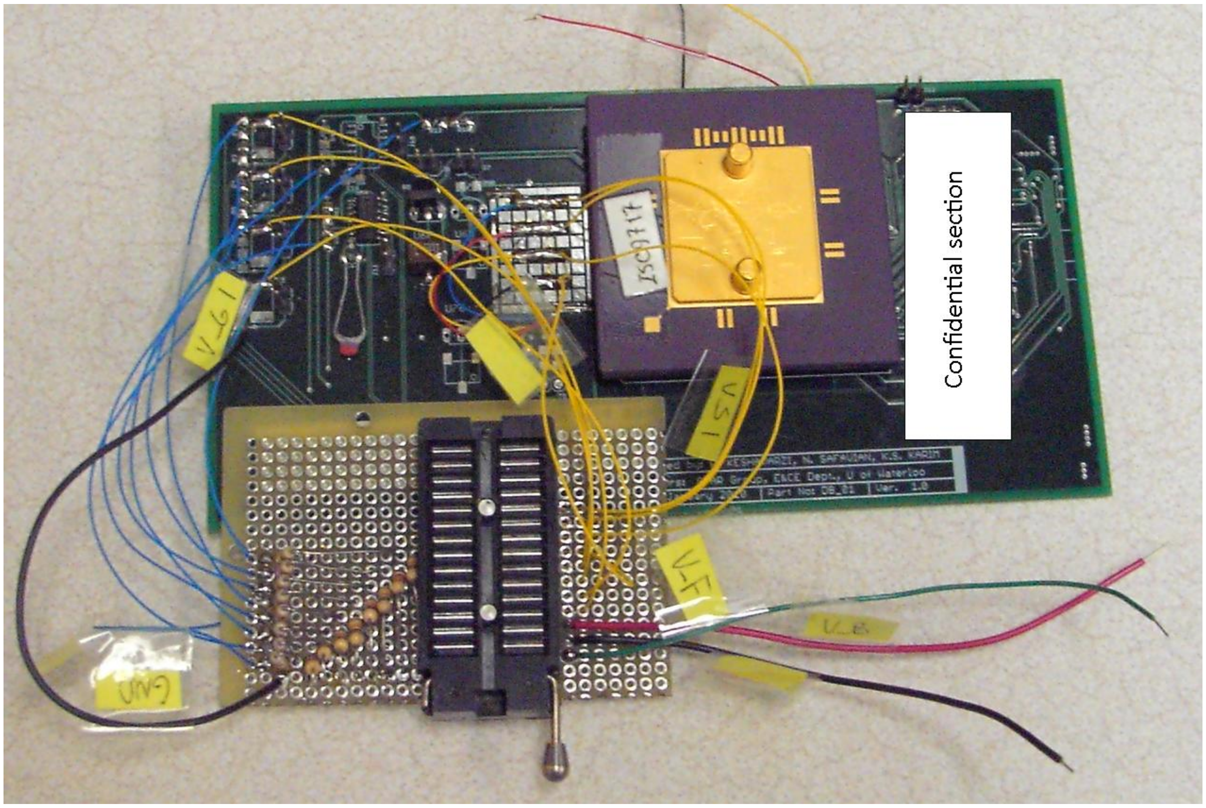


Figure 4.8 – Daughter-board PCB for characterization of Indigo chip

4.3 Captured frames for 32x32 PPS array

A 32x32 PPS array was manufactured in G2N lab by STAR group members. It was then wire-bonded to a daughter-board. Therefore 32 inputs out of the 64 available inputs on the daughter-board are wire-bonded this array, and the other 32 input are not connected.

Before any attempt to capture a frame using the interface, another categorization of the inputs to Indigo chip is necessary. Figure 4.9 shows the columns sorted for this case.

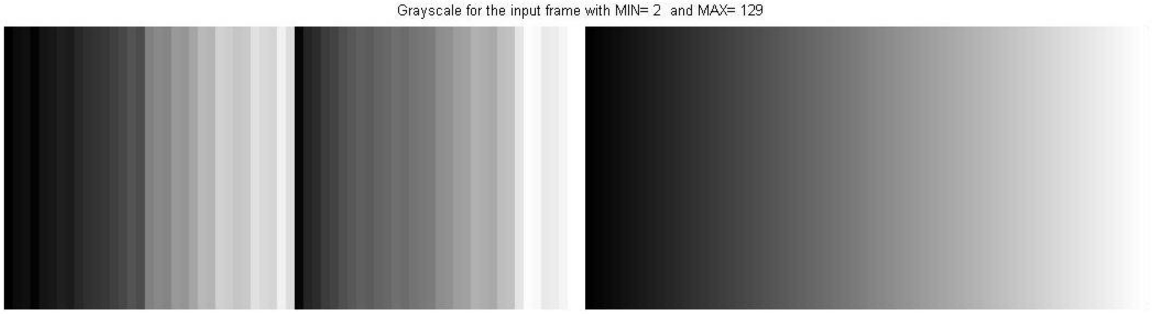


Figure 4.9 – Pattern generated for categorized inputs

In Figure 4.9, the right-side half (64 columns) are the inputs of the Indigo that are not wire-bonded to the PGA package. But the left-side half (64 columns) has also two parts. The very left-side quarter (32 columns), are those inputs that are wire-bonded to the 32x32 PPS array. The centre quarter (32 columns), are the inputs on the daughter-board that are not wire-bonded to the 32x32 PPS array. The central part is for the inputs that are wire-bonded to PGA package and therefore are present on the daughter-board, but are not used in this case where the array size is only 32x32.

Figure 4.10 shows several frames captured by the 32x32 PPS array when the array is diagonally covered by an aluminum sheet.

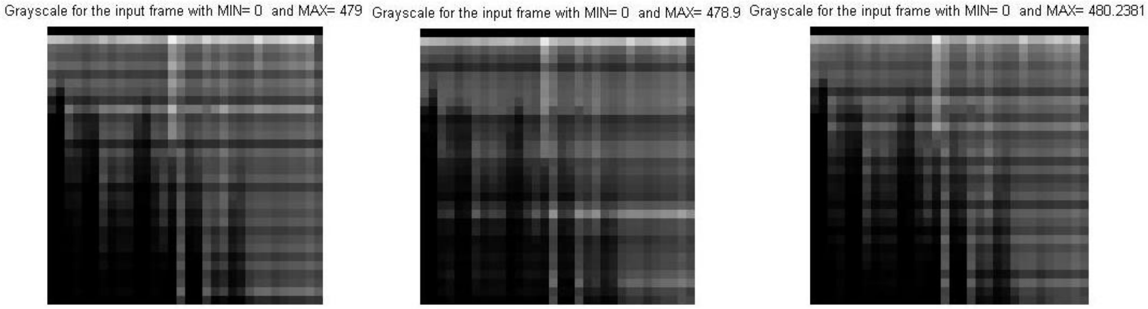


Figure 4.10 – Captured frames from PPS array half-covered diagonally by aluminum sheet

It is for the case where PCB in Figure 4.7 is mounted on the main PCB in Figure 4.6, and the array is half-covered by a blocking layer as show in Figure 4.11.

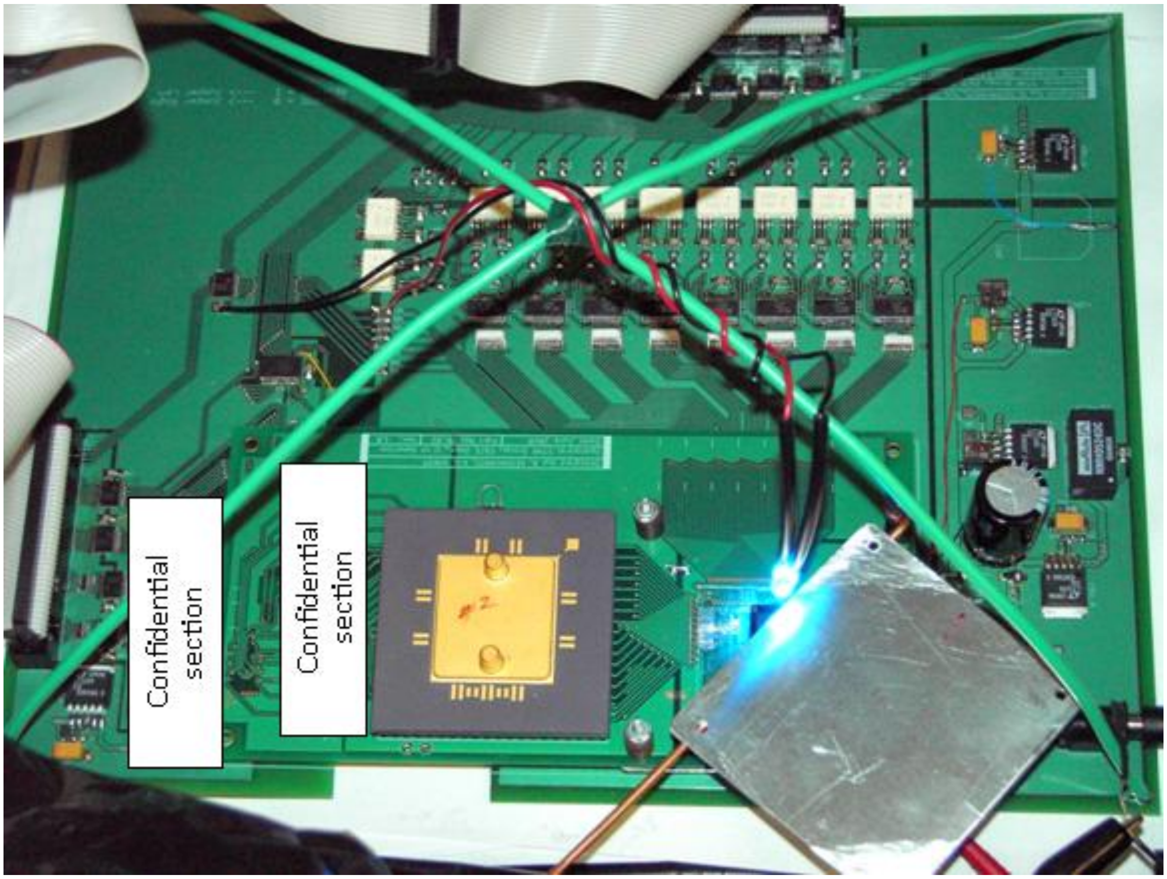


Figure 4.11 – PPS array half-covered by aluminum sheet diagonally

The light source in the ideal case would be an X-ray generator. But for the purpose of testing this prototype interface, a blue color LED with spectrum close to X-ray band was used.

A close look at Figure 4.10 may pose this question that why the edge of the frame generated is not sharp. Several factors may contribute to this effect. First, the edge of the aluminum sheet is not smooth, and therefore it causes different diffractions along the edge. Second, there are reflections inside the chamber where the test is conducted. Third, there is non-uniformity on pixels on the array due to process variations. This non-uniformity results in non-uniform sensitivity to light. Some suggestions to improve this feature are given in the next chapter.

Chapter 5 Conclusion and recommendations

This chapter evaluates the interface functionality against initial goals upon which the interface was designed. It also proposes some suggestions for improvement in the interface's functionality, for the case that this work would be continued and expanded in future.

5.1 Contributions

The following is a short list of achievements in this work:

- The Indigo chip was employed for a panel readout circuitry
- A main PCB designed to host the main blocks of the circuit
- A daughter-board PCB designed to characterize the Indigo chip
- A VHDL code developed to interface the main PCB to the personal computer (PC)
- A MATLAB program developed to capture and show the frames sent by the FPGA
- The functionality of the whole system was tested and confirmed by capturing a few sample frames

All the above points were achieved for the first time in the STAR group for X-ray panel frame capture.

5.2 Conclusion

By comparing the above contributions with the initial goals of this work, which were described in section 2.1, the following conclusions can be made.

5.2.1 Hardware

There is only one main PCB and one daughter-board. The daughter-board should be customized for each type of array such as PPS, APS, etc. The structure of the hardware is relatively simple, straight forward, and easy to understand. So, it will not be difficult to train other STAR group members to work with this interface.

Maximum width of arrays can be up to 64 pixels. Original goal was 128 pixels which is the same as the number of inputs to Indigo chip. Due to difficulty of wire-bonding of the Indigo die to its PGA package, we had to cut-down this feature. A 64x64 array would have over 4000 pixels and is capable of showing decent patterns.

5.2.2 Serial port

Data is sent from FPGA to PC over a serial cable in the current version. Due to slow speed of RS-232 protocol, it takes several seconds to transmit the information for one frame. It is quite acceptable for digital still X-ray imaging. But if the goal shifts towards real-time X-ray imaging, then this bottle-neck would pose a limit on the interface's capability.

5.2.3 MATLAB interface

The current version of the interface uses MATLAB software on the PC in order to receive data through the serial port and display the captured frame on the monitor graphically. Even though MATLAB is a great tool for prototype development, it is not the best tool to produce the final product in software world. It is very processing power demanding and runs relatively slow. Besides, it does not create an executable file similar to compilers. This is a drawback of the current version of the interface.

5.2.4 Power supply

As described in section 3.3.9 this interface is using a single-pole power supply to energize the interface. The fact that a dual-pole power supply is not required when using this interface, makes it much easier and faster to work with this interface.

5.2.5 Overall functionality

Considering all the above facts, it is quite fair to conclude that the interface meets the majority of its initial goals. The interface is capable to capture frames up to 64 pixels in width. The speed of frame-grabbing is okay. It takes 3 to 5 seconds to capture one frame from a 32x32 array, which is acceptable for a prototype sample.

5.3 Recommendations

Depending on what performance measure is expected to be maximized, there are several ways to improve the interface functionality in future works. They are discussed in the following sections.

5.3.1 Array size

The maximum width of the array can be increased from 64 pixels to 128 pixels while one Indigo chip is employed. But the way Indigo die is currently wire-bonded to the PGA package would have to be changed. Two ways can be considered for this change:

- Using same or other type of package but with a better wire-bonding diagram
- Using a different approach as opposed to package. For example, flip-chip method

For the first approach, it may be quite feasible to do it. But it would be very costly to do so for a low quantity prototype job. Wire-bonder machines can be programmed to move the needle in a certain path inside the cavity of the package. Since it is a three dimensional routing, the task is time-consuming and therefore expensive.

For the second approach, new ways of employing the Indigo chip should be examined, for example flip-chip method can be investigated. In this method a metal pattern similar to that of the Indigo chip should be developed on top of a wafer material, like glass. The Indigo chip is then mounted upside-down on top of this pattern to make connections between corresponding pads.

5.3.2 Speed of operation

The RS-232 protocol can provide a maximum speed of 115,200 bps. This would normally be considered enough for digital still X-ray imaging. However for higher-rate of refreshing, a higher-rate of data communication would be required. USB is a perfect channel when data rates are in the range of mega-bits or even mega-bytes per second. But handling data communication is more challenging from a software development point of view.

5.3.3 User interface

The MATLAB interface is slow to run. A faster interface can be written using other programming languages. C, C++, and C# are examples of good programming languages that are suitable for this interface. They provide the capability to work at low level with hardware and bit-wise programming which is necessary for serial port handling, and at the same time they are powerful in graphical representations. Besides, since they are compiler software, they create executable files for running under the operating system. This last feature makes the interface much faster, even fast enough for real-time applications.

5.3.4 Miscellaneous

Here are a few small suggestions that would make working with the interface a bit easier:

5.3.4.1 Power-on LED

Adding an LED, for example Green color, to distinguish when the main PCB is powered up from the case that the main PCB is not powered would provide a user with a quick way to realize whether the board is powered-up or not.

5.3.4.2 Indigo-chip on main-board

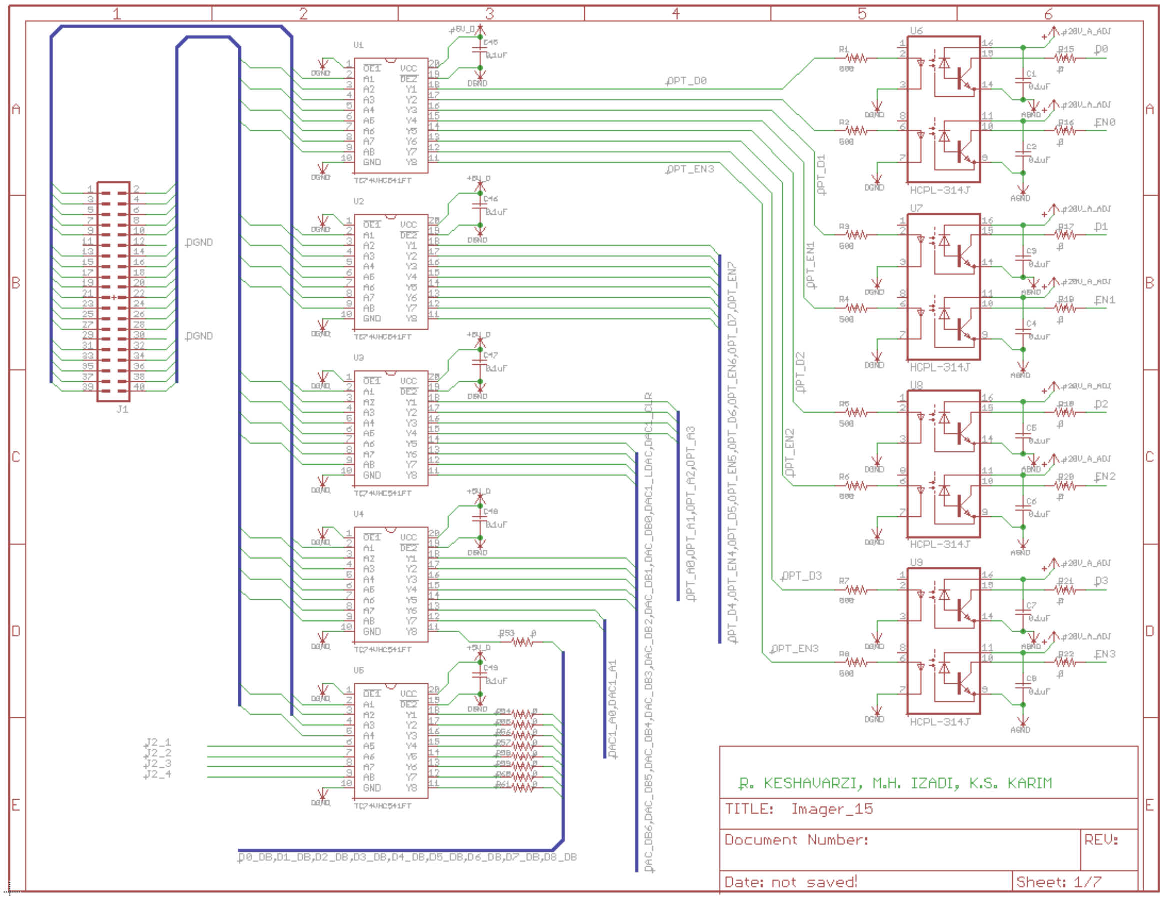
In the current version of the interface, the Indigo chip is located on the daughter-board. Therefore its distance to at array is minimum, which is good from noise point of view.

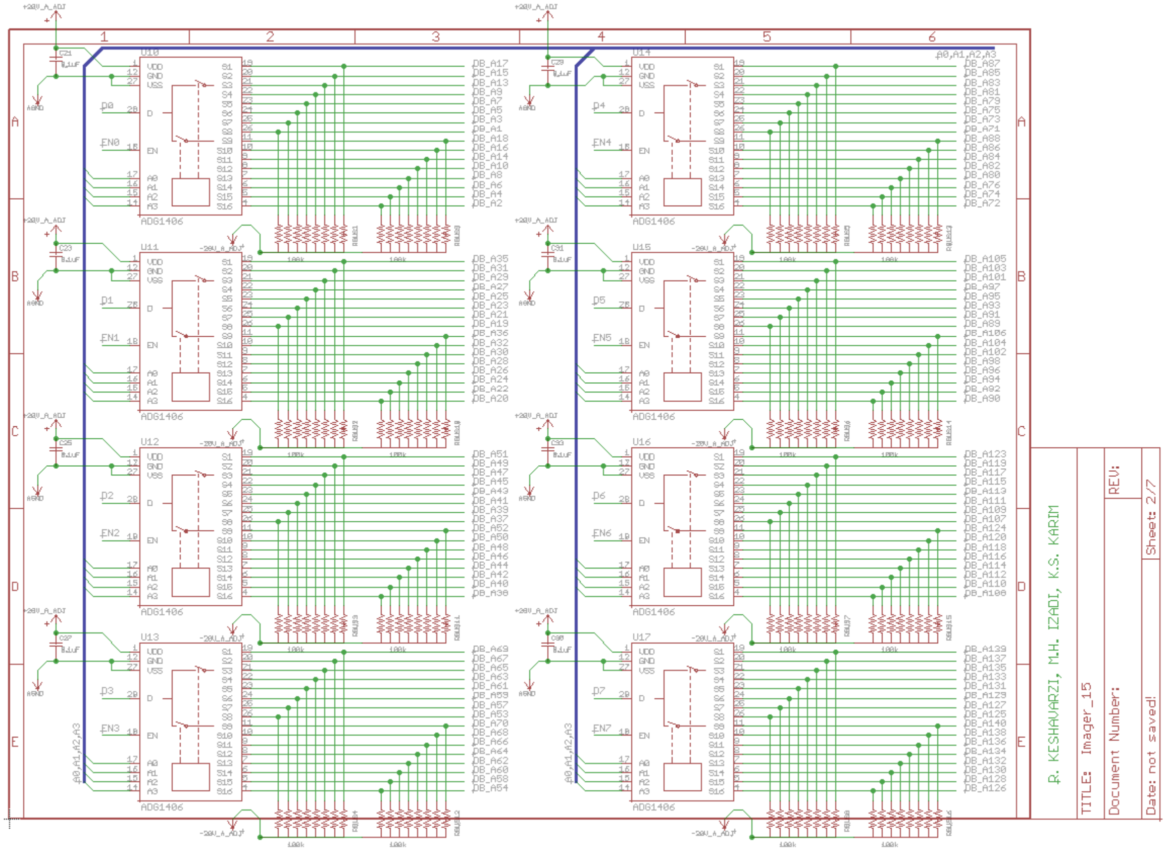
If noise minimization is not the goal in an array test and frame capture, then the Indigo can be moved to main-board. In this case, the distance between Indigo and array would be longer. Besides, a connector would be inserted between the two components. Therefore parasitic capacitance associated with array outputs would increase in this case. However, in return, the need to have one PGA socket and one Indigo chip for each array would be eliminated. This is a great saving from three points of views of cost, time, and ease of work. The cost of PGA socket and Indigo chip for each daughter-board is significant. And it takes time to solder a PGA socket with 455 pins to a PCB for each daughter-board. Finally, it makes wire-bonding of the array to daughter-board much easier. With the PGA socket soldered on the daughter-board, a special fixture should be used in order to facilitate wire-bonding of the array to daughter-board.

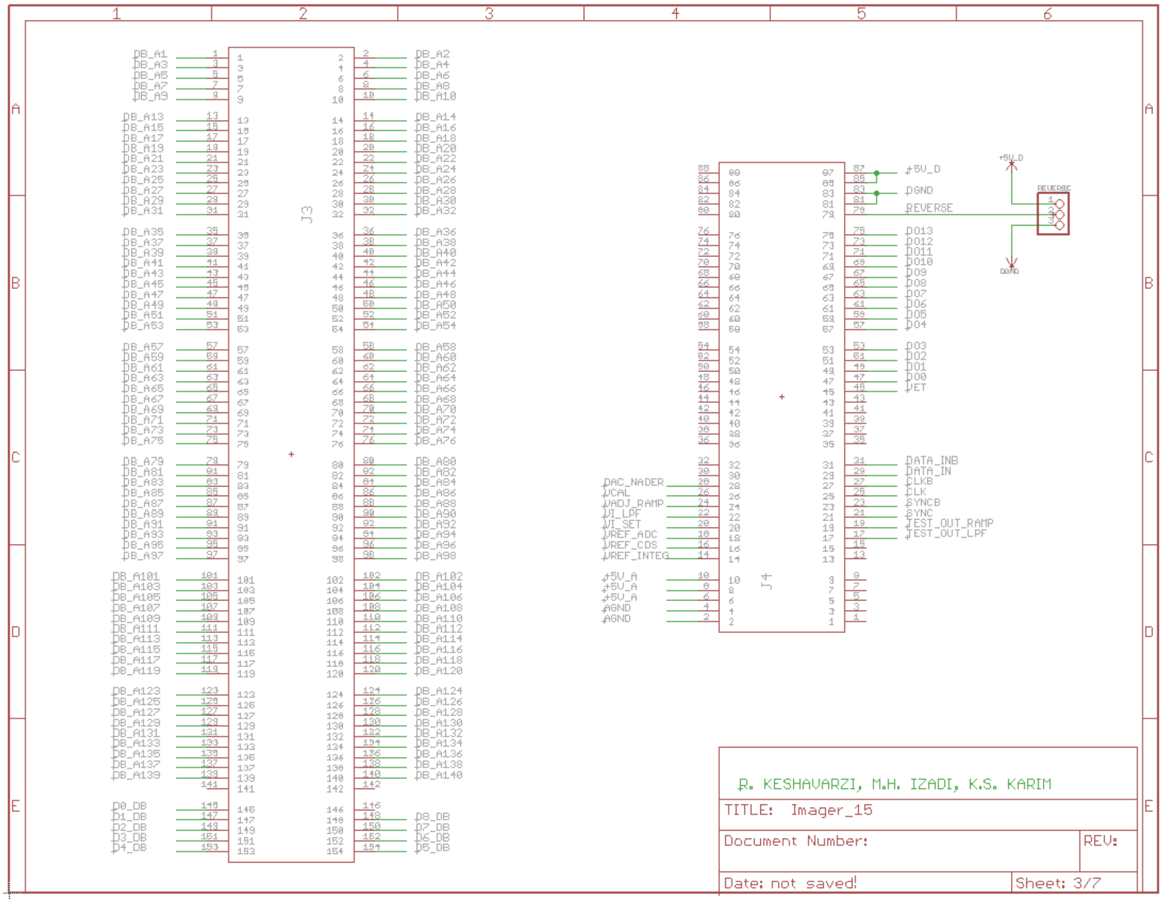
As such, putting the Indigo chip and its socket on the main-board as opposed to daughter-board, has many advantages except for noise level in the system. However, since the added noise due to putting the Indigo on the main PCB is a fixed-pattern noise, it can be eliminated using mathematical calculations and software.

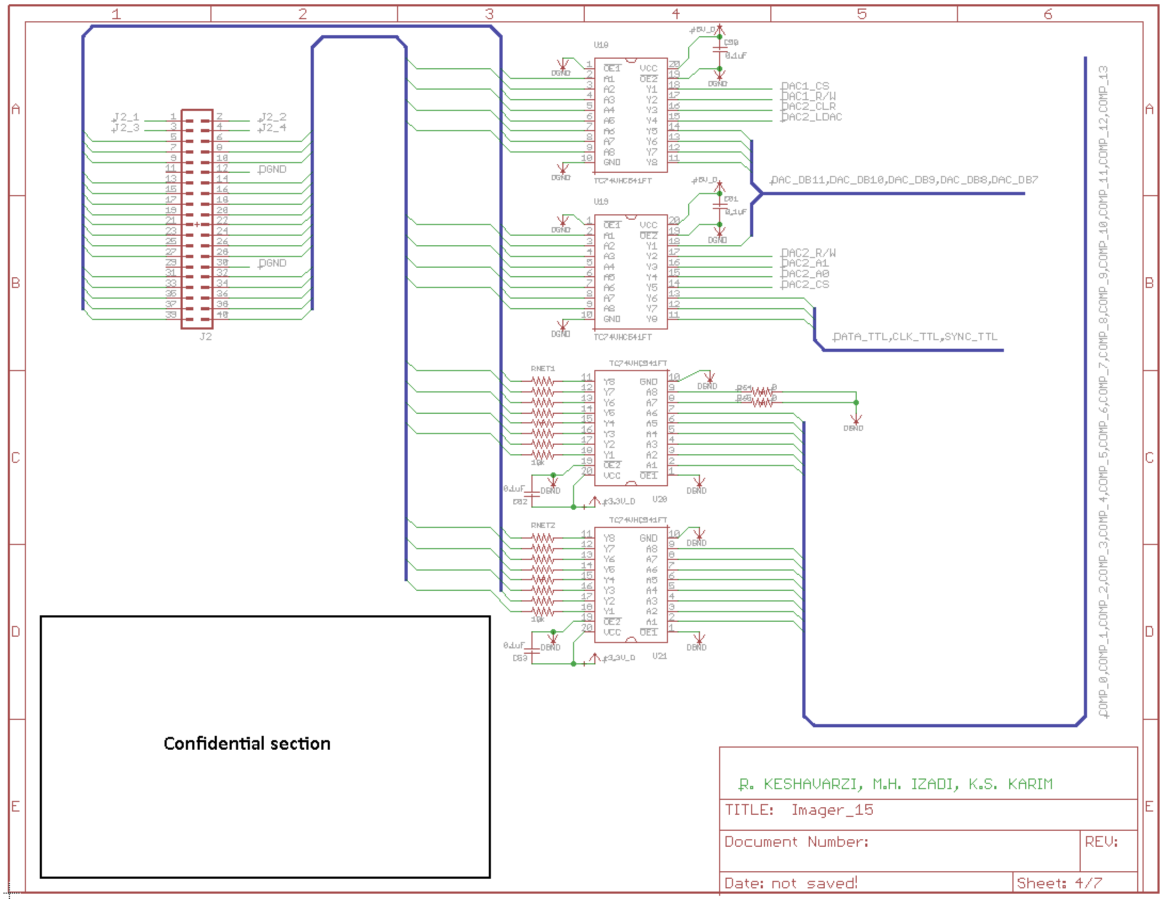
Appendices

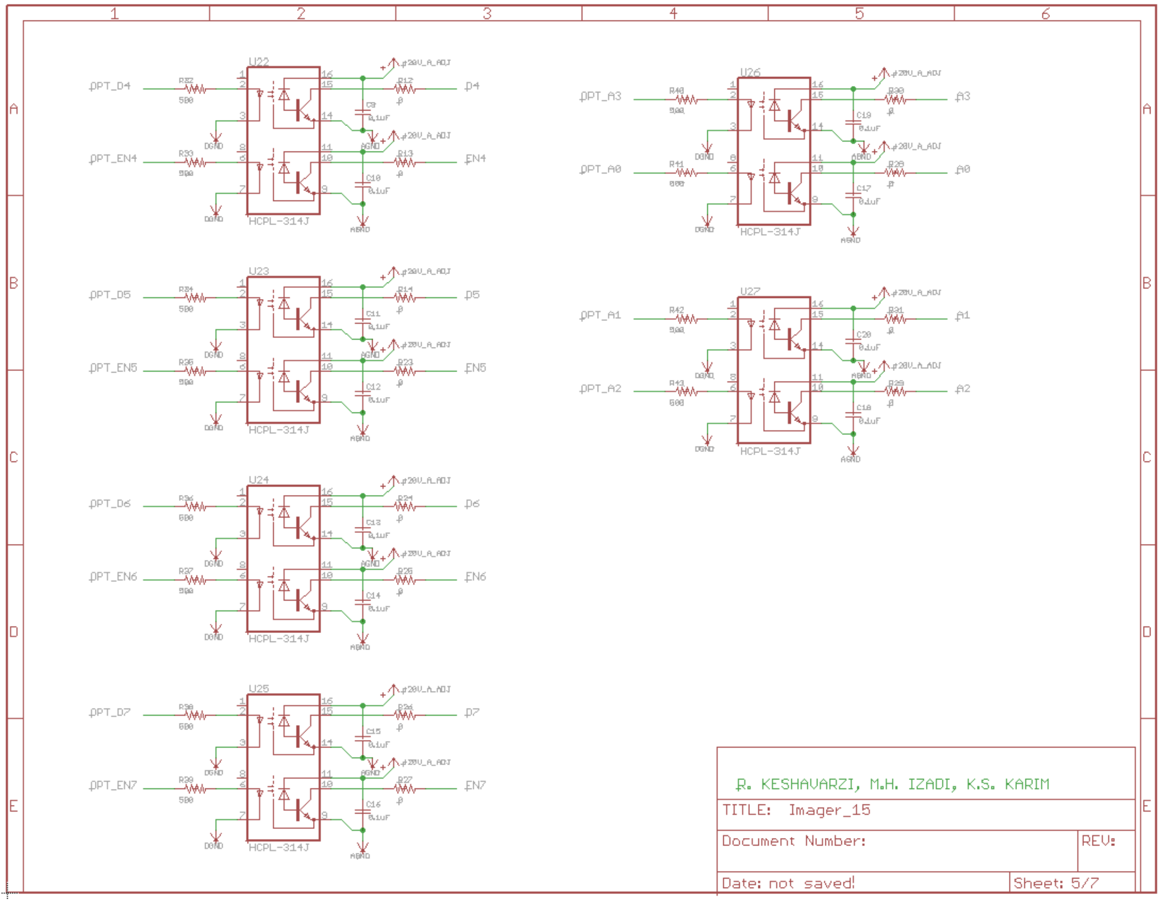
Appendix A – Schematic diagrams of the main board




R. KESHAVARZI, M.H. IZADI, K.S. KARIM
TITLE: Imager_15
Document Number:
REV:
Sheet: 2/7
Date: not saved!
U10
U11
U12
U13
U14
U15
U16
U17
ADG1406
+20V_A_ADJ
-20V_A_ADJ
AGND
A0,A1,A2,A3
D0
D1
D2
D3
D4
D5
D6
D7
EN0
EN1
EN2
EN3
EN4
EN5
EN6
EN7
100k
0.1uF

1
2
3
4
5
6
A
B
C
D
E
J3
DB_A1
DB_A3
DB_A5
DB_A7
DB_A9
DB_A13
DB_A15
DB_A17
DB_A19
DB_A21
DB_A23
DB_A25
DB_A27
DB_A29
DB_A31
DB_A35
DB_A37
DB_A39
DB_A41
DB_A43
DB_A45
DB_A47
DB_A49
DB_A51
DB_A53
DB_A57
DB_A59
DB_A61
DB_A63
DB_A65
DB_A67
DB_A69
DB_A71
DB_A73
DB_A75
DB_A79
DB_A81
DB_A83
DB_A85
DB_A87
DB_A89
DB_A91
DB_A93
DB_A95
DB_A97
DB_A101
DB_A103
DB_A105
DB_A107
DB_A109
DB_A111
DB_A113
DB_A115
DB_A117
DB_A119
DB_A123
DB_A125
DB_A127
DB_A129
DB_A131
DB_A133
DB_A135
DB_A137
DB_A139
D0_DB
D1_DB
D2_DB
D3_DB
D4_DB
DB_A2
DB_A4
DB_A6
DB_A8
DB_A10
DB_A14
DB_A16
DB_A18
DB_A20
DB_A22
DB_A24
DB_A26
DB_A28
DB_A30
DB_A32
DB_A36
DB_A38
DB_A40
DB_A42
DB_A44
DB_A46
DB_A48
DB_A50
DB_A52
DB_A54
DB_A58
DB_A60
DB_A62
DB_A64
DB_A66
DB_A68
DB_A70
DB_A72
DB_A74
DB_A76
DB_A80
DB_A82
DB_A84
DB_A86
DB_A88
DB_A90
DB_A92
DB_A94
DB_A96
DB_A98
DB_A102
DB_A104
DB_A106
DB_A108
DB_A110
DB_A112
DB_A114
DB_A116
DB_A118
DB_A120
DB_A124
DB_A126
DB_A128
DB_A130
DB_A132
DB_A134
DB_A136
DB_A138
DB_A140
D8_DB
D7_DB
D6_DB
D5_DB
J4
+5V_D
DGND
REVERSE
D013
D012
D011
D010
D09
D08
D07
D06
D05
D04
D03
D02
D01
D00
VET
DATA_INB
DATA_IN
CLKB
CLK
SYNCB
SYNC
TEST_OUT_RAMP
TEST_OUT_LPF
DAC_NADER
VCAL
VADJ_RAMP
VI_LPF
VI_SET
VREF_ADC
VREF_CDS
VREF_INTEG
+5V_A
+5V_A
+5V_A
AGND
AGND
+5V_D
REVERSE
DGND
R. KESHAVARZI, M.H. IZADI, K.S. KARIM
TITLE: Imager_15
Document Number:
REV:
Date: not saved!
Sheet: 3/7


J2
J2_1
J2_2
J2_3
J2_4
DGND
DGND
U18
U19
U20
U21
TC74VHC541FT
C30
C31
C62
C63
0.1uF
+5V_D
+3.3V_D
DGND
DAC1_CS
DAC1_R/W
DAC2_CLR
DAC2_LDAC
DAC_DB11,DAC_DB10,DAC_DB9,DAC_DB8,DAC_DB7
DAC2_R/W
DAC2_A1
DAC2_A0
DAC2_CS
DATA_TTL,CLK_TTL,SYNC_TTL
RNET1
RNET2
10k
R64
R65
COMP_0,COMP_1,COMP_2,COMP_3,COMP_4,COMP_5,COMP_6,COMP_7,COMP_8,COMP_9,COMP_10,COMP_11,COMP_12,COMP_13
Confidential section
R. KESHAVARZI, M.H. IZADI, K.S. KARIM
TITLE: Imager_15
Document Number:
REV:
Date: not saved!
Sheet: 4/7

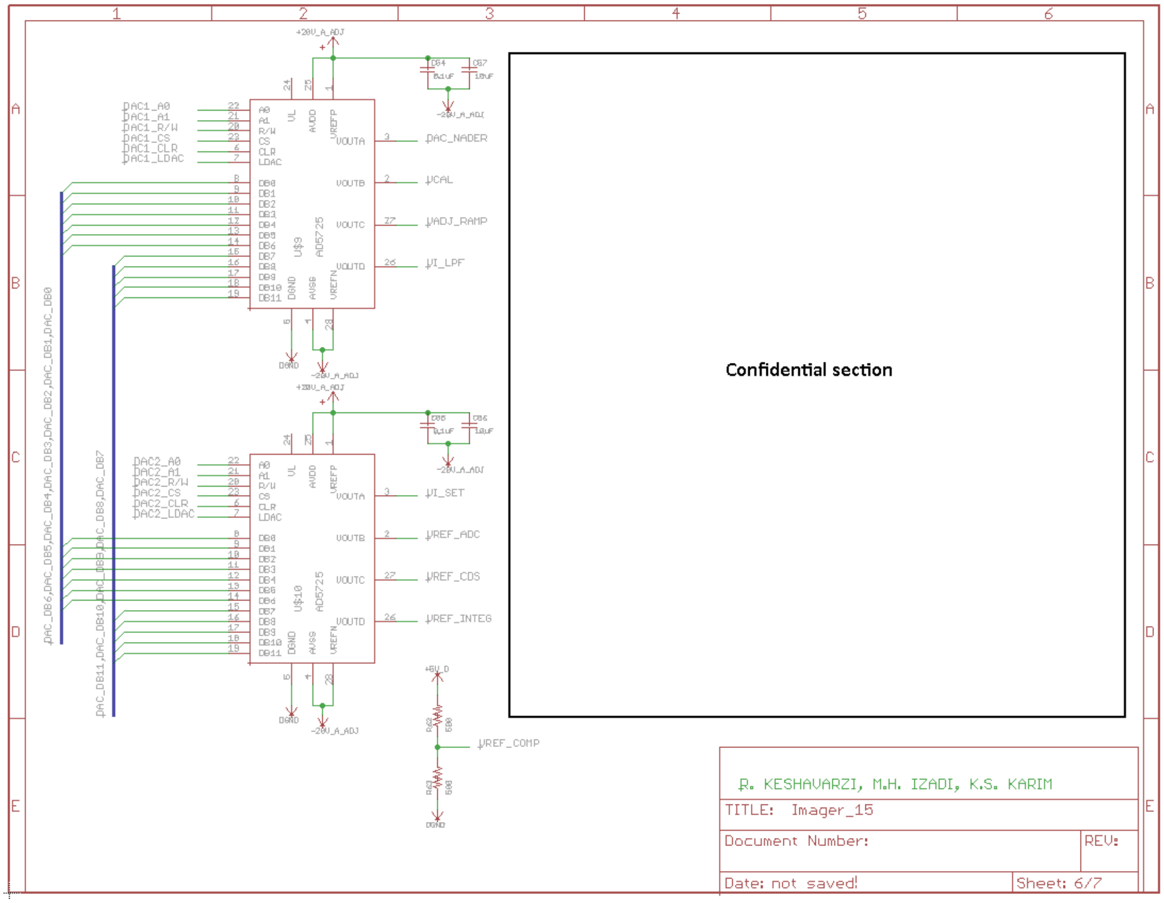


R. KESHAVARZI, M.H. IZADI, K.S. KARIM

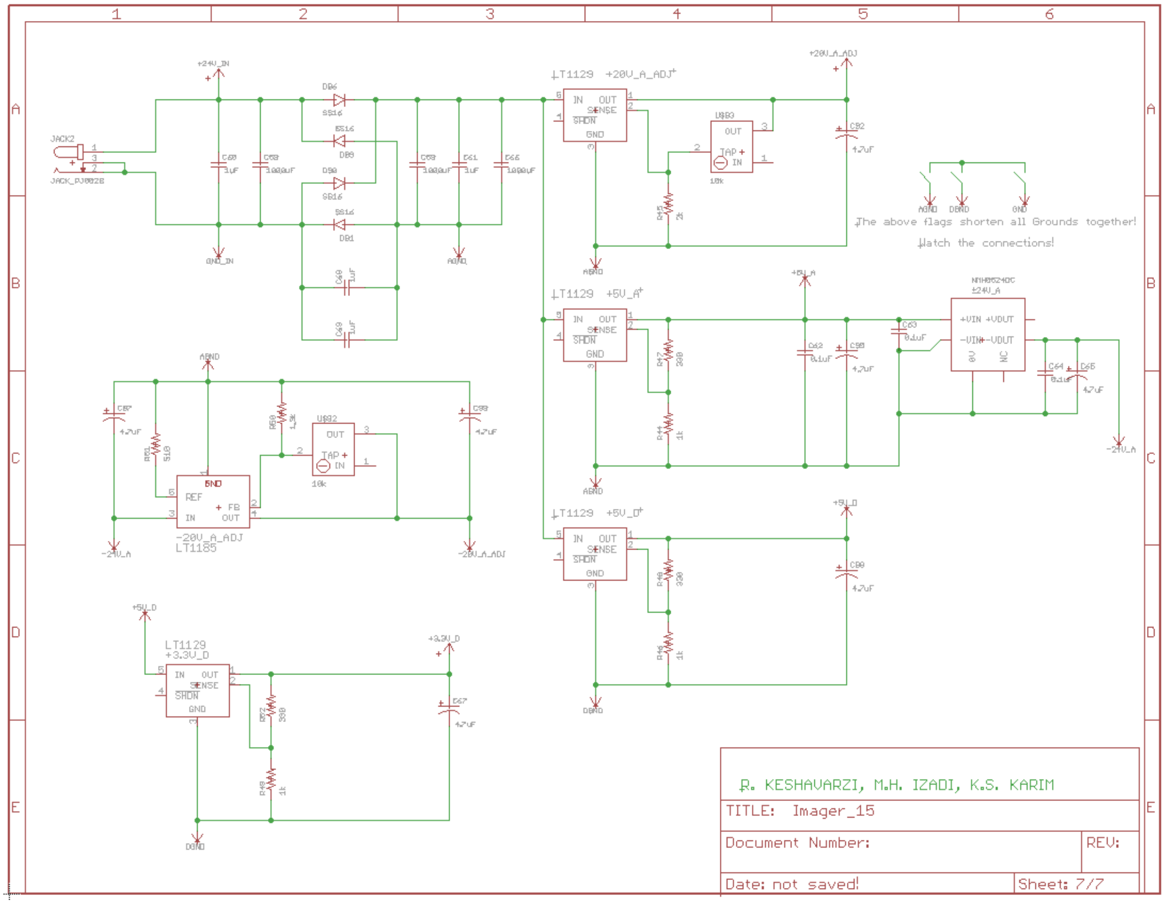
TITLE: Imager_15

Document Number: REV:

Date: not saved! Sheet: 5/7

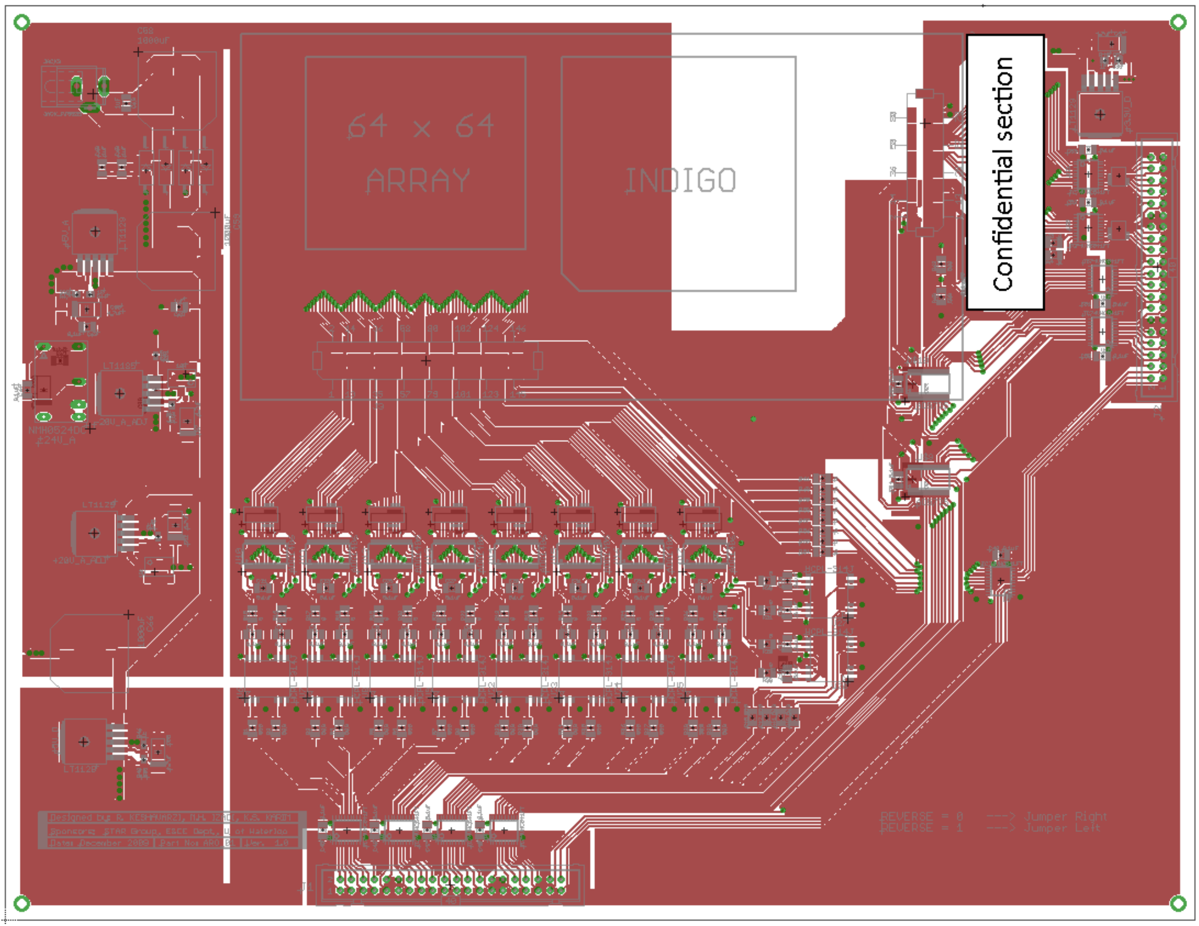


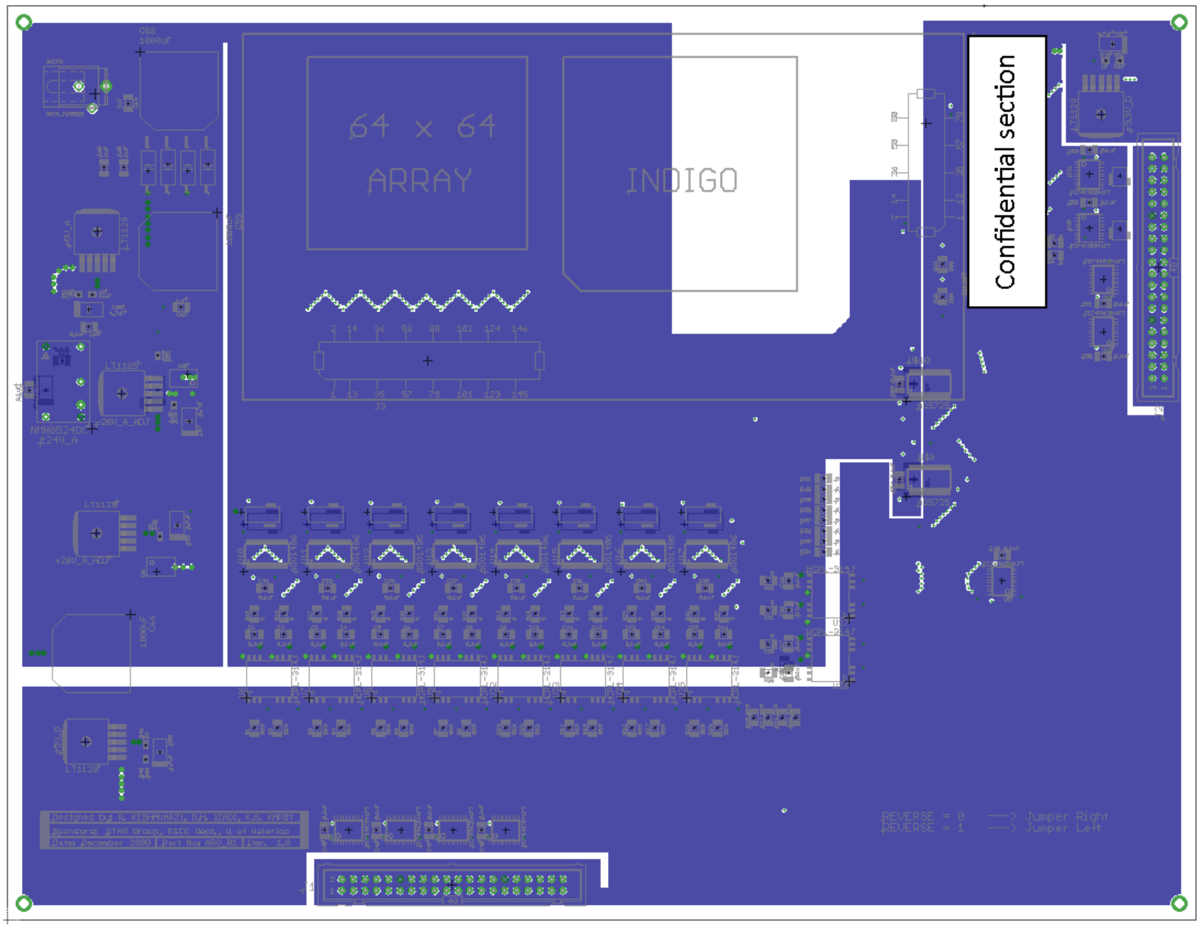
+20V_A_ADJ
C54
0.1uF
C67
10uF
-20V_A_ADJ
DAC1_A0
DAC1_A1
DAC1_R/W
DAC1_CS
DAC1_CLR
DAC1_LDAC
A0
A1
R/W
CS
CLR
LDAC
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DB8
DB9
DB10
DB11
VL
AVDD
VREFP
VOUTA
VOUTB
VOUTC
VOUTD
DGND
AVSS
VREFN
U$9
AD5725
DAC_NADER
VCAL
VADJ_RAMP
VI_LPF
DGND
-20V_A_ADJ
+20V_A_ADJ
C85
0.1uF
C86
10uF
-20V_A_ADJ
DAC2_A0
DAC2_A1
DAC2_R/W
DAC2_CS
DAC2_CLR
DAC2_LDAC
U$10
AD5725
VI_SET
VREF_ADC
VREF_CDS
VREF_INTEG
DGND
-20V_A_ADJ
DAC_DB6,DAC_DB5,DAC_DB4,DAC_DB3,DAC_DB2,DAC_DB1,DAC_DB0
DAC_DB11,DAC_DB10,DAC_DB9,DAC_DB8,DAC_DB7
+5V_D
R62
500
VREF_COMP
R63
500
DGND
Confidential section
R. KESHAVARZI, M.H. IZADI, K.S. KARIM
TITLE: Imager_15
Document Number:
REV:
Date: not saved!
Sheet: 6/7

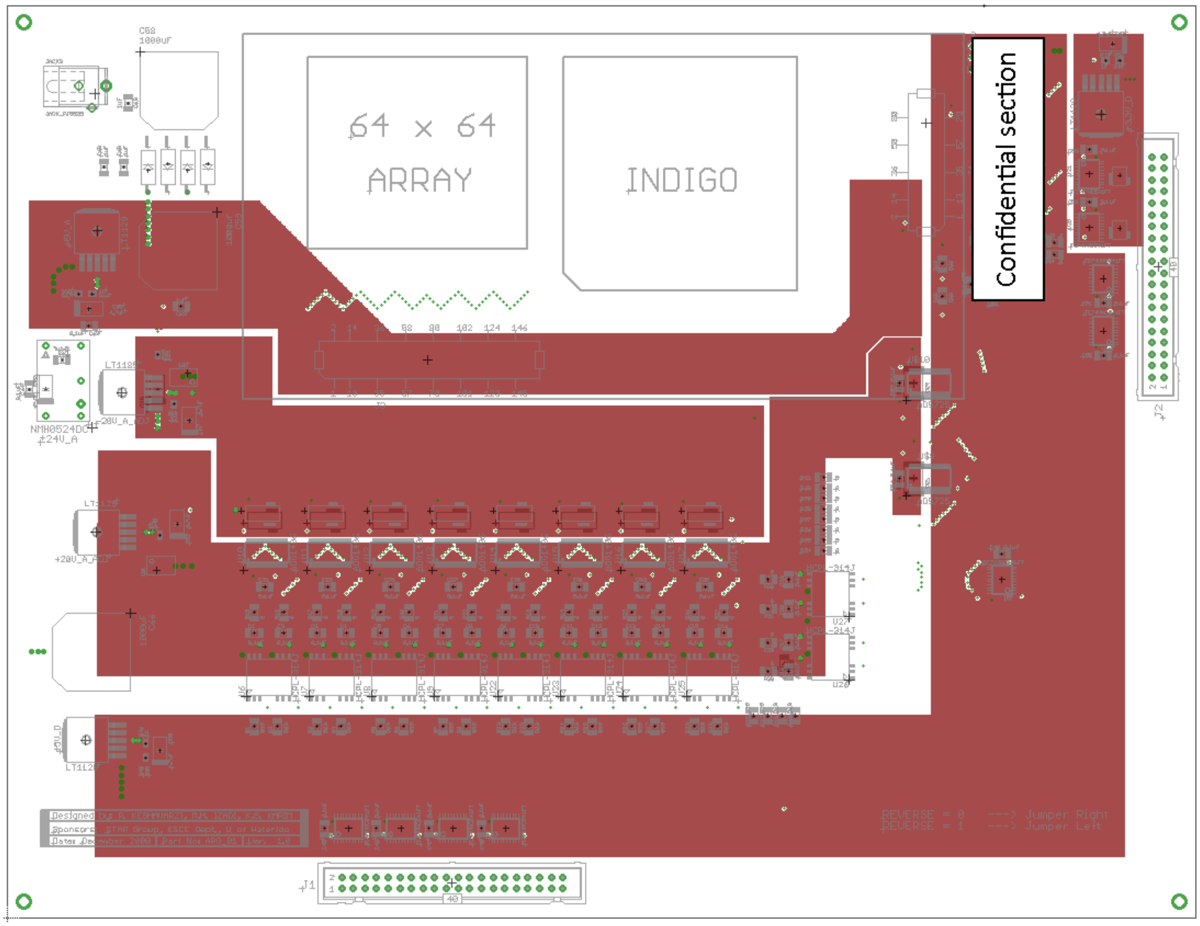


JACK2
JACK_PJ002B
+24V_IN
GND_IN
D06
SS16
D08
D90
D01
C60
1uF
C68
1000uF
C69
C61
C66
C40
C49
AGND
LT1129 +20V_A_ADJ
IN
OUT
SENSE
SHDN
GND
U$03
TAP
10k
R45
2k
C92
4.7uF
+20V_A_ADJ
AGND
DGND
GND
The above flags shorten all Grounds together!
Watch the connections!
LT1129 +5V_A
R47
200
R44
1k
+5V_A
C62
0.1uF
C90
C63
NMH0524DC
±24V_A
+VIN
+VOUT
-VIN
-VOUT
0V
NC
C64
C65
-24V_A
C87
R61
510
R60
1.3k
U$02
REF
FB
IN
OUT
-20V_A_ADJ
LT1185
C99
-24V_A
-20V_A_ADJ
LT1129 +5V_D
R48
330
R46
1k
+5V_D
C80
DGND
+5V_D
LT1129
+3.3V_D
R27
330
R10
1k
+3.3V_D
C67
DGND
R. KESHAVARZI, M.H. IZADI, K.S. KARIM
TITLE: Imager_15
Document Number:
REV:
Date: not saved!
Sheet: 7/7

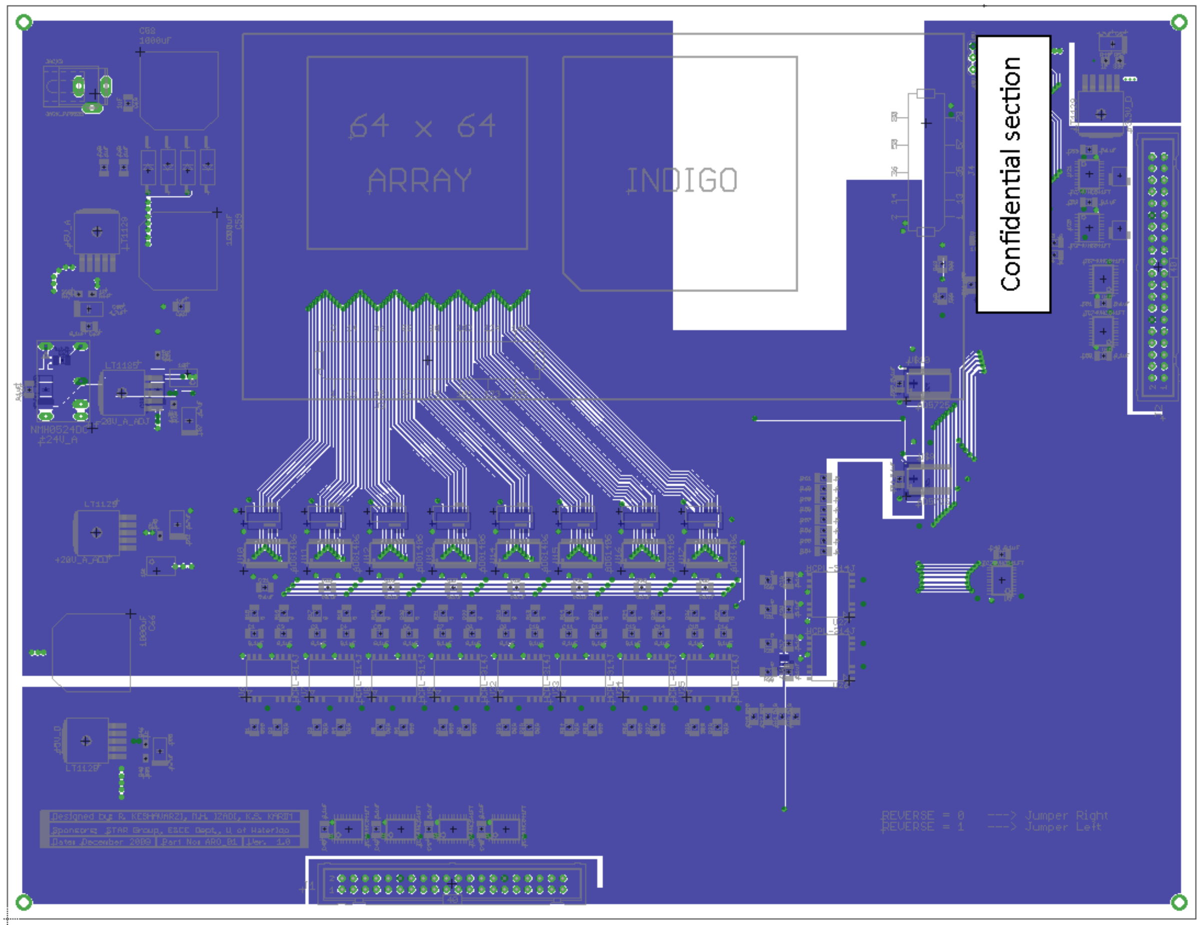
Appendix B – Layouts for the main board


64 x 64
ARRAY
INDIGO
Confidential section
LT1128
Designed by: R. KESHAVARZI, M.H. IZADI, K.S. KARIM
Sponsors: STAR Group, E&CE Dept., U of Waterloo
Date: December 2009
Part No: ARO_B1
Ver. 1.0
REVERSE = 0 ---> Jumper Right
REVERSE = 1 ---> Jumper Left


64 x 64
ARRAY
INDIGO
Confidential section
NMH0524DC
LT1128
LT1129
+20V_A_ADJ
REVERSE = 0 ---> Jumper Right
REVERSE = 1 ---> Jumper Left
Designed by: R. KESHAVARZI, M.H. IZADI, K.S. KARIM
Sponsors: STAR Group, E&CE Dept., U of Waterloo
Date: December 2009
Part No: ARO_01
Ver. 1.0



C68
1000uF
64 x 64
ARRAY
INDIGO
Confidential section
LT1129
LT1121
NMH0524DC
±24V_A
+20V_A_ADJ
LT1129
+20V_A_ADJ
+5V_D
LT1128
J1
J2
40
Designed by: R. KESHAVARZI, M.H. IZADI, K.S. KARIM
Sponsors: STAR Group, E&CE Dept., U of Waterloo
Date: December 2008
Part No: ARO_01
Ver. 1.0
REVERSE = 0 ---> Jumper Right
REVERSE = 1 ---> Jumper Left



64 x 64
ARRAY
INDIGO
Confidential section
C66
1000uF
LT1129
NMH0524DC
+24V_A
+20V_A_ADJ
LT1129
+20V_A_ADJ
LT1128
J1
40
REVERSE = 0 ---> Jumper Right
REVERSE = 1 ---> Jumper Left
Designed by: R. KESHAVARZI, M.H. IZADI, K.S. KARIM
Sponsors: STAR Group, E&CE Dept., U of Waterloo
Date: December 2009 | Part No: ARO_01 | Ver. 1.0

References

[1] W. W. Morgan, P. C. Keenan, E. Kellman, **An atlas of stellar spectra, with an outline of spectral classification**, Chicago, IL, The University of Chicago press, 1943

[2] **On A New Kind of Rays**, Röntgen, Wilhelm Conrad**.** 274, Würzburg : Nature, 1896, Vol. 53

[3] **"*The Generation of X-Rays*"**, Available at epswww.unm.edu/xrd/xrdclass/03-Gen**Xrays**.ppt(Accessed July 30, 2012)

[4] T. T. Tan, H. S. Sim, S. P. Laua, H. Y. Yang, M. Tanemura, J. Tanaka, **"*X-ray generation using carbon-nanofiber-based flexible field emitters*"**, Applied Physics Letters **88**, 103105 (2006)

[5] Hristovski, Christos, **Design and Characterization of an 8x8 Lateral Detector Array for Digital X-Ray Imaging**, M.A.Sc. Thesis, Waterloo, ON, University of Waterloo, 2010

[6] ***"X-Ray Hand"***, Available at http://x-ray-spine.x-rayss.com/x-ray-hand (Accessed July 30, 2012)

[7] J. A. Seibert ***"Flat-panel detectors: how much better are they?"***, Pediatric Radiology, pp 173-181, Volume 36, Supplement 2 (2006)

[8] ***"ISC9717_V2: ROIC for Instrumentation and X-ray Detector"***, Indigo® Datasheet, http://www.flir.com/uploadedFiles/9717spec.pdf (Accessed July 30, 2012)

[9] ***"Quad, 12-Bit, Parallel Input, Unipolar/Bipolar, Voltage Output DAC"***, AD5727 Datasheet, http://www.analog.com/static/imported-files/data_sheets/AD5725.pdf (Accessed July 30, 2012)

[10] ***"Micropower Low Dropout Regulators with Shutdown"***, LT1129 Datasheet, http://cds.linear.com/docs/Datasheet/112935ff.pdf (Accessed July 30, 2012)

[11] ***"Low Dropout Regulator"***, LT1185 Datasheet, http://cds.linear.com/docs/Datasheet/1185ff.pdf (Accessed July 30, 2012)

[12] ***"Micropower Precision Low Dropout Series Voltage Reference Family"***, LT1461 Datasheet, http://cds.linear.com/docs/Datasheet/1461f.pdf (Accessed July 30, 2012)

[13] **Development and Education Board**, Altera DE2 Board User Manual, http://cds.linear.com/docs/Datasheet/1461f.pdf (Accessed July 30, 2012)

[14] **R2012a Documentation → Image Processing Toolbox**, MATLAB Online Manual, http://www.mathworks.com/help/toolbox/images/ref/imshow.html (Accessed July 30, 2012)