# Ramp ADC with hyperbolic compression for high dynamic range image sensor in sequential 3D CMOS

Simen Stjernen

Thesis submitted for the degree of
Master in Elektronikk, informatikk og teknologi
60 credits

Department of physics
Det matematisk-naturvitenskapelige fakultet

UNIVERSITY OF OSLO

Autumn 2021

# Ramp ADC with hyperbolic compression for high dynamic range image sensor in sequential 3D CMOS

Simen Stjernen

Ramp ADC with hyperbolic compression for high dynamic range image
sensor in sequential 3D CMOS

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

# Abstract

In this project I have designed a high dynamic range image sensor with a 3D CMOS technology process development kit (PDK). The idea is to build the whole circuit in pixel (ADC and counter). Each pixel contains its own ADC, and the complete image sensor would consist of an array of these pixels and peripheral logic to read the individual outputs of these ADCs.The 8 bit ADC is a modification of a ramp ADC resulting in a non-linear projection of light intensity as input onto a digital output to achieve HDR.

My motivation for picking this particular project was two-fold 3D CMOS technology is an exciting development and has large space saving potential. In this project I employ an experimental 3D technology called 3D sequential integration, which can be characterized by a 3D interconnect density of close to $1^8$ 3D contacts per $mm^2$, about 100 times denser than commercially available technologies. This allows for very dense 3D CMOS designs in a volume since stacking is permitted. A full HDR ADC can fit on the footprint of a $4um^2$ photo pixel with 5 total CMOS tiers[2]. One such tier would be exclusively reserved for photo diodes, allowing for a 100% fill factor, since the entire surface area would be sensing light and still have in-pixel processing elements.

The thesis builds on former work done by Mikkel Mikkelsen[3] and Phillip Häfliger[2]. Mikkel built a similar circuitry in a 0.35 um 2S CMOS technology, with a 15% fill factor. Phillips paper[2] discusses the theory of non-linear HDR, including hyperbolic compression. In this project I am consentrating on the practical implementation in 3D sequential integration, comparing it to the theory and providing some analysis on the robustness of the physical implementation.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 HDR

Providing a High Dynamic Range is extremely important in the production of image sensors. It makes us capable of recreating images of the real world close to what we see. Without it images would be very dark or light. Providing a result close to what we see with the human eye. First follows a introduction to the terms Dynamic and High Dynamic Range. Then a section on how we can improve the DR using different techniques and a section on different projections of the IO characteristics.

### 1.1.1 Dynamic Range

Dynamic range is defined as the ration between the max and the min tonal value in an image. Said plainly the darkest and the lightest part. Improving the dynamic range towards low light is often discussed as improving sensitivity, as we are able to detect smaller differences in light intensity. However high sensitivity and a high dynamic range are usually in a trade of relationship. We would like to achieve both. Standard digital imaging techniques only offer a certain differtial between the lightest of light and the darkest of dark. If we want to see clear differences in the very light or the very dark sections of a single image and thus increase the dynamic range (HDR), we have to apply other techniques [7].

One such technique is combining several narrow range exposures of the same image. We narrow down the dynamic range of many images and then combine them. To see clear details in a dark image we need high exposure and for very bright situations we need a low exposure image. This is used in multiple exposure HDR sensors.

Some HDR sensors only use a single exposure process. This is used to mitigate the time latencies caused by the difference in exposure time in multiple exposure HDR sensors. If the subject is fast moving in a high contrast environment this can prove useful.

In this project I create a multiple exposures HDR sensor. In every pixel the intensity of the light coming in is compared to a pre set ramp as it happens. Each pixel will therefore have a slightly different exposure time, since the input will cross the ramp at different times giving different outputs. Combining all the outputs will hopefully give a higher dynamic range than we would otherwise get.

One massive advantage of building such circuitry in 3D-CMOS technology is that we can fit more pixels on the same area as it would have taken up before since we can place counters and such under the pixels. More pixels will give the oppurtunity to pick out more details in the light and dark sections, as we can combine a greater range of images like in Fig 1.1

The combination of images in fig 1.1 can be seen as a piecewise linear projection. By using dual or multiple exposure HDR image sensors it is possible to make a extremely good projection even at low or bright light. The higher the number of sensors the more accurate. Rather than a overarching projection of the entire range, smaller section are put together to portray smaller parts of a eventually bigger picture. This is however only one possible way to non lineary project an image. Some other possibilities are covered in section 1.1.2.

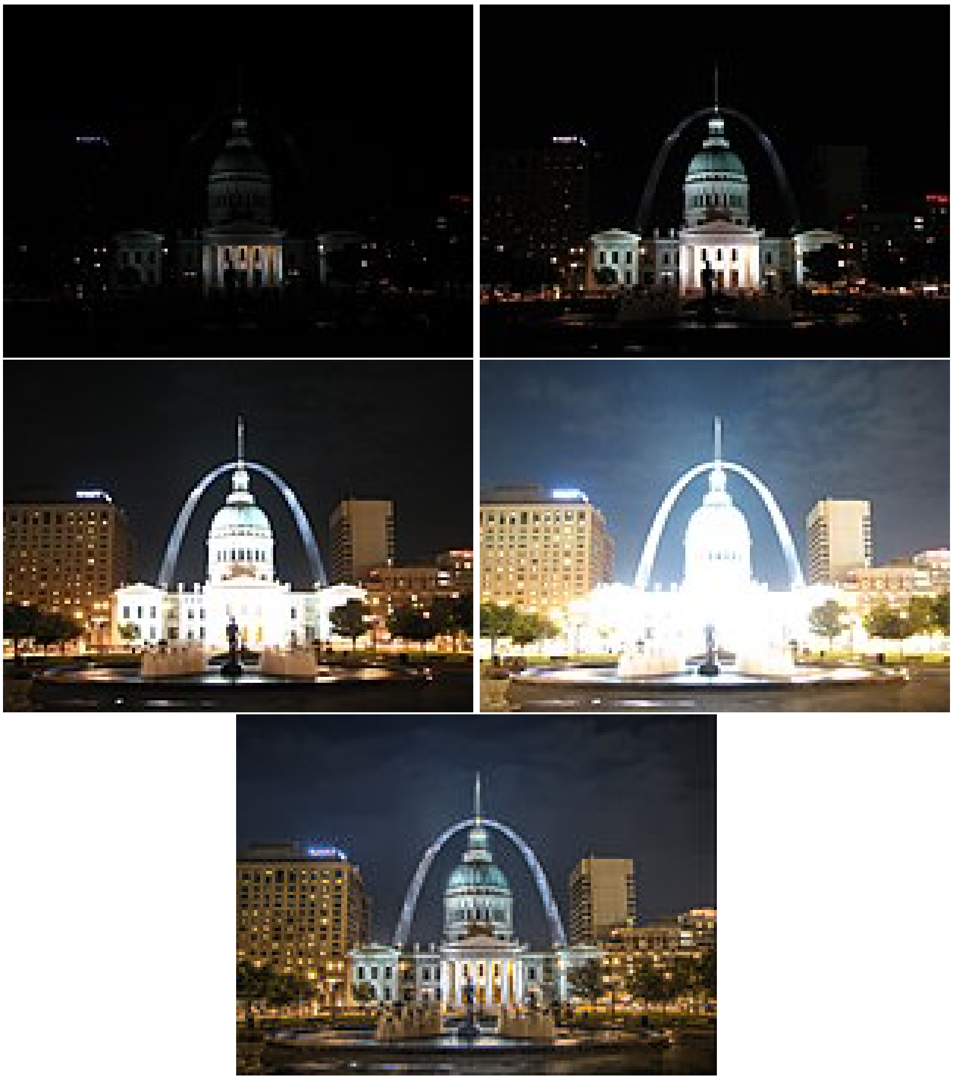


Figure 1.1: Four images of varying exposure (-4, -2, 2, 4) and the local tone mapping of the four images [7]

### 1.1.2 IO Characteristics

IO relationship, or In Out relatioship is the quite simply how the input affects the output of whatever we are measuring. It is a good way of visualizing what is actually happening in the circuitry and information on how we should further process the data.

A piecewise linear approach was covered in 1.1.1, but we may also want non linear projections. They will however all be convex.

It can give a multitude of results including linear, hyperbolic or logarithmic projection depending on how we portray it.

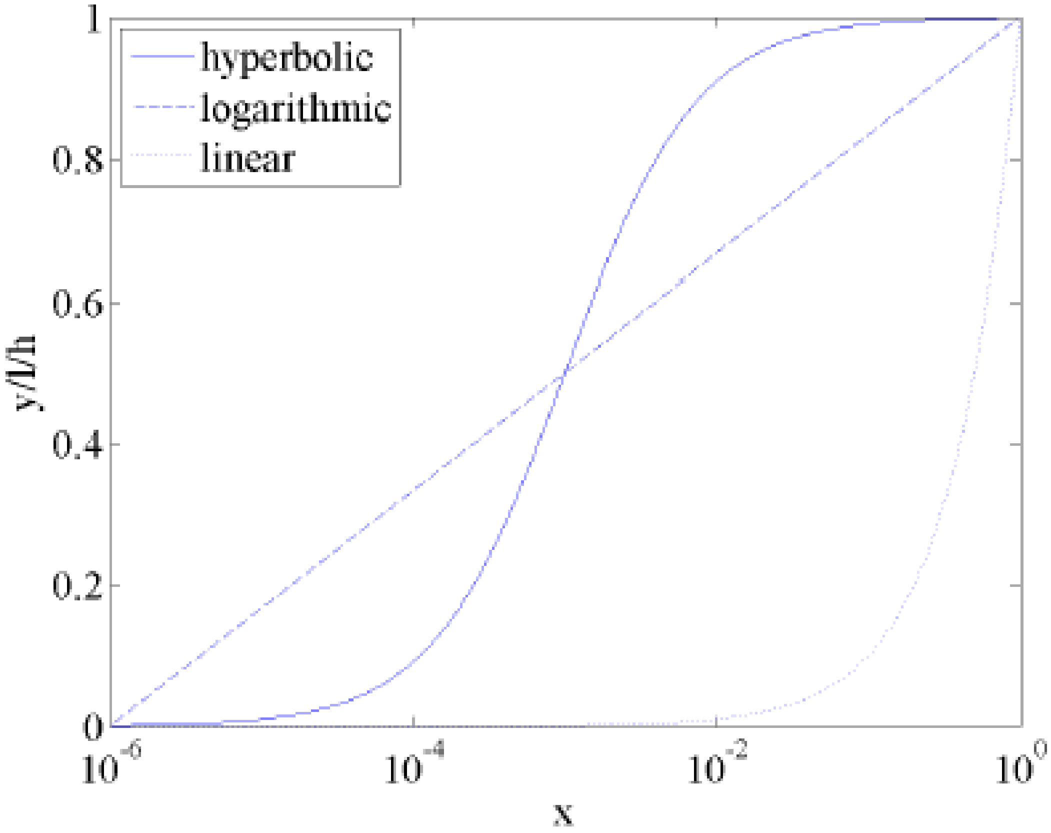


Figure 1.2: Possible ways of portraying the IO relationship [3]

In this project these two elements are given by the current derived from the foto sensor and the corresponding voltage the comparator gives (count). Since I would like to represent a high dynamic range, the very light and very dark is especially interesting. What will a small current do to the IO realtionship. The image sensor in the project will produce a hyberbolic IO relationship. This is then provable following this equation from the paper [2].

$$h(x) : [0, \infty] \to [0, 1], h(x) = \frac{mx}{mx + 1}$$

Which is the compressive projection which resulst in a hyperbola, as seen in Fig 1.2 a big part of the range is occupied by small and high voltages (y/l/h). Changes in x around these point give small changes in y/l/h, thus we get a lot of images around this voltage with different exposure times.

## 1.2 CMOS

The most popular process today for building p and n-type MOSFETS is found in CMOS technology. Complementary metal-oxide-semiconductor is a type of transistor fabrication process that uses symmetrical and complementary pairs of p and n-type MOSFETS for logic functions. It is used for constructing a range of integrated circuits such as microprocessors, microcontrollers, memory chips and othe digital logic (inverters, NAND, AND, OR...), it is also used in analog circuits such as image sensors, data converters, RF circuits and others. Two important characteristics of CMOS devices are high noise immunity and low static power consumption.[6] The rest of this section covers some circuitry that can be built using CMOS that are important to this project.

### 1.2.1 Photo diode

Photodiodes are semi-conducting devices that converts photons (light) into electrical current. The simplest photodiode is a simple PN-diode.The P-layer has an abundance of positive charge(holes), whereas N has overload of negative charge. In the depletion region between the layers no free cariers exists, this creates an electric field across it which only allows current flow in one direction. [4]

If the PN-junction is exposed to light, the photons if they have enough energy can cause covalent bonds to break, forming electron-hole pairs in the depletion region. The electric field sweeps the electrons to the n side (positive cathode) and the holes to the p side (negative anode), giving rise to a reverse current across the junction. The current is proportional to the intensity of the light received. Light has been converted to an electric signal.[5. Sedra & Smith (2016), p. 224]

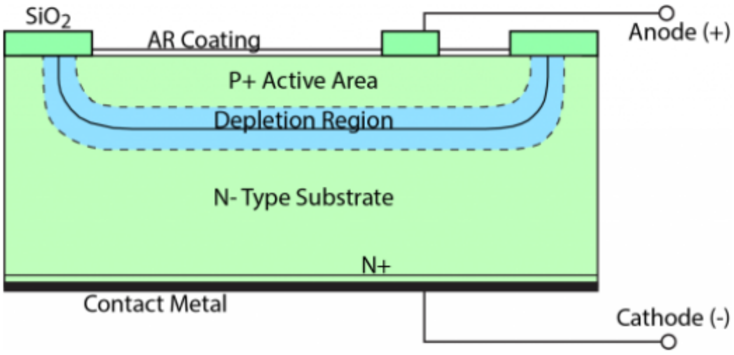


Figure 1.3: Possible PN diode[4]

### 1.2.2 ADC

An ADC converts a analog signal to a digital one using transistors, there are many different subsections of ADC convertert including flash, dual slope, pipeline and ramp.

An ADC converts a continuous time and amplitude analog signal to a discrete time and amplitude digital signal. The conversion is performed periodically, sampling the input rather than continuously.

In this project I use a modified ramp ADC which compares a value coming in with a preset ramp and switches when the ramp is greater than signal in, the time measured until the switch occurs is the digital value of the input signal. In this project this signifies that the sensor has picked up a signal.

### 1.2.3 Flip-Flop

A flip-flop is a simple circuit capable of storing a single bit of information. They are one of the basic storage elements in sequential logic. They have two possible states 0 and 1, hence the name as the can change between the two when triggered by an input signal. Flip-flops can be level-triggered (latch) or edge triggered [1]

### 1.2.4 Counter

Flip-flops are an important part of many larger componenets, one such component are counters.

Counters are used to quantify how many times a certain incident occures over a period of clock cycles, we split these into two main categories syncronous and non syncronous.

In a syncronous counter every element is simultaneously triggered by an external clock. All the flip-flops change value at the same time in parallel. In a non-syncronous counter on the other hand any element is dependent on the one before it, apart from the first element which is clocked. The signal will be delayed passing through as the flip-flops get triggered sequentialy. Because of this behavior we call these ripple-counters.[11]

In this project I use a syncronous counter. (256 bit) The numbers of bits signifies how high the counter can count $2^n$. Included in the system is also a reset witch takes the number of bits to reset the counter n.

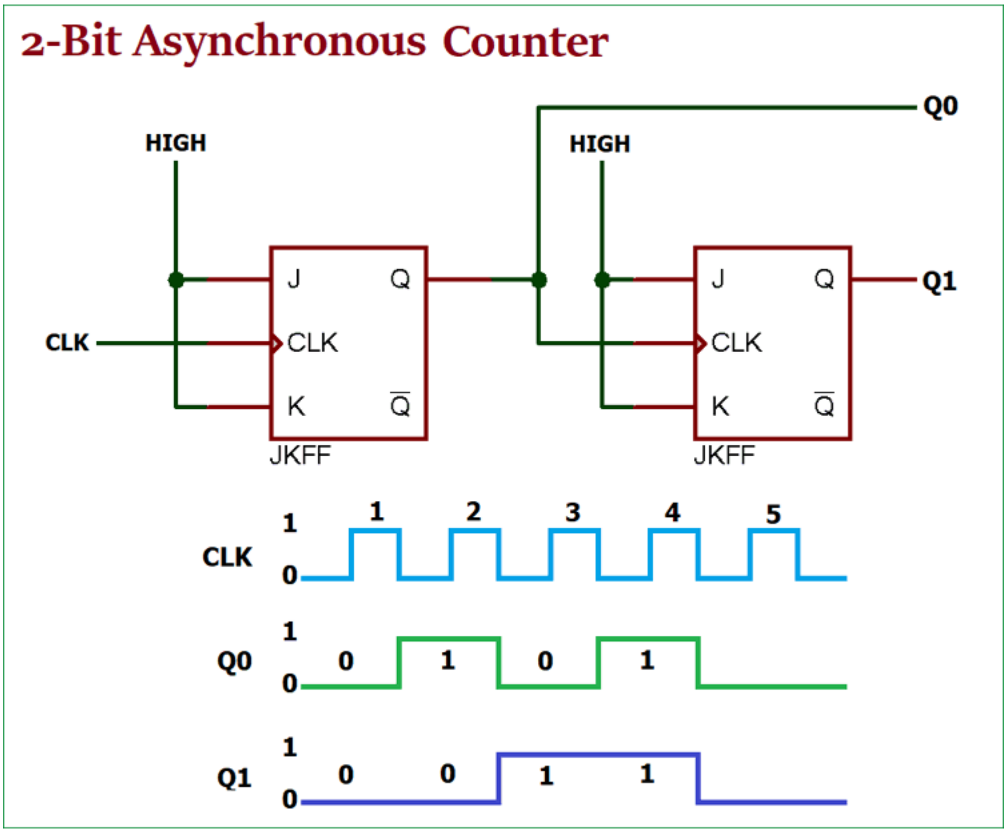


Figure 1.4: Behavior of a JK Asyncronous counter[10]

## 1.3 3D-CMOS

3D-CMOS is quite simply building IC in multiple layers, either in stack or in cube (sequential integration) both described in subsection 1.3.5. 3D CMOS makes it possible to combine different CMOS processes in a single system. Below follows a few benefits of building in 3D-CMOS.

### 1.3.1 State of the art

In modern 3D circuit building and in CMOS parallel integration, the wafers are processed separately and then bonded together later. In contrast 3DSI bonds the unprocessed wafer and then makes transistors on it with high precision and small 3D connections.3D in this context applies to the usage of top and bottom tier wafer and transistors. These can be of different sizes and either contain digital or analog circuitry. This is one of the massive benefits. We can build the digital and the analog parts seperatly using the most effective tecniques known and merge them later.

### 1.3.2 Size

We can fit much more functionality into a small area. This expands on Moores law and enable the production of small but powerfull circuits. For exapmle in image sensor circuit, ideally we would like to cover 100% of the surface area with photo diodes. With 3D-CMOS we would be able to achieve this by building layers with different capabilities together and putting counter, read out and such beneath the photo diode arrays.

### 1.3.3 Cost

The partion of a larger chip into multiple parts for stacking, can improve profits as parts can be tested induvidialy and sizes can vary. Different wafers can be used for different layers, cutting cost.

### 1.3.4 Integration

Parts can be built by different processes, on different wafers and are therefore easier to optimize to the conditions and applications we need. Components that would otherwise be incompatible (production, wafers) can be combined.

### 1.3.5 3D-MUSE

3D-MUSE is a EU-project trying to stimulate progress from system-in-stack to true system-in-cube integration made possible by 3D sequential integration as mentioned above. By developing and working on this technology in state if the art 28 nm CMOS will enable 3D integrated circuit design where the the inter-tier vertical connections via density is of the same order of magnitude as 2D horizontal wire density. [13]

### 1.3.6 System in stack vs. System in Cube (Sequential integration)

System in stack is characterized by locating functional blocks within a single plane in the 3D integration stack. Blocks are built on separate wafers, stacked and connected by 'face-to-face' stacking or copper copper bonding through silikon vias. Established 3D integration has a connection density in the vertical between tiers that is one or several orders of magnitude smaller than the connection density in 2D CMOS technology [13]. Typically there are only a few large through silicon vias connecting tiers in the z-direction.[13]

System in cube makes full use of the freeing up space in the interconnect density in the third dimension of sequential 3D integration and implements functional blocks across multiple tiers. 3D sequential integration paves the way for vertical connection density on par with horizontal connection density. Mixed signal blocks can now be spread in a volume [13]

# Part I

# The project

The project is based on a former thesis by Mikkel Mikkelsen. In it he built a fotosensor using a ramp ADC built out of a comparator and a counter to get a digital reading. He does this using 2D CMOS technology. My assignment was to make a similar circuitry, but ported to 3D using Leti. The project is simulation only as there was not enough time to have it manufactured, and such technology is still at an early stage.

The project consists of building a CMOS image sensor with an ADC per pixel placed beneath the sensor in a 3D build. The ADC is a ramp ADC witch compares the intensity of the light it receives with a preset ramp. The comparison happens simultaneously in such a way that we can get varying exposure times based on the cross point of intensity of light and the ramp. Giving us a high dynamic range on the output.

We will seek to prove this by plotting and interpreting the IO relationship of the circuitry, and assuring that it gives a result were happy with, and that falls in line with the counter values

Building in 3D is relatively new technology and gives us big advantages in saving of space, protection of digital circuitry (bottom) and usage of mixed circuitry. It is a perfect for for this project as we have a analog sensor on top of a digital counter and converter. A finished sensor would consist of a series of these components.

By building in 3D you should be able to get a high dynamic range sensor on a smaller area than you would in 2D. We can fit more photo diode arrays on an area than we otherwise would.

# Chapter 2

# Building and Testing

## 2.1 Full circuit

I will start by showing of the full circuit, and going into details of individual parts further down

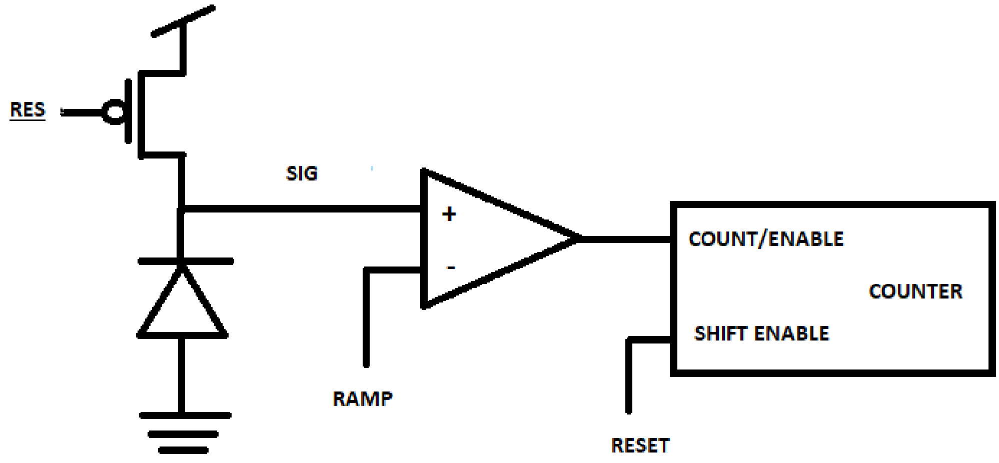


Figure 2.1: The whole circuit block diagram

The full circuit compares the input obtained from a sensor (SIG) with a predecided ramp (RAMP).SIG is a photoelectric integrated on the parasitic capacitance of the node. For a constant photostream in the exposure time this will produce a falling ramp with a gradient proportional to the photo current. This is proved in the next section. RAMP is a rising ramp which goes from Gnd to Vdd within the exposure time. The counter keeps track of how long it takes for the two to meet through the COUNT/ENABLE input. The value obtained from the counter will therefore give us a non-linear projection monotonically falling as the intensity of the light increases. There is also reset/shift functionality on both the sensor input (RES) and the counter elements (SHIFT/ENABLE) that resets.

## 2.2 ADC

The first stage was building and testing the comparator using 28 nm transitors from the CCLETI library in Cadence. This is the same comparator Mikkel used in his thesis [3] which he obtained here [9, p.463].

I have switched the transistors out for top tier 28nm ones from Leti and added a slight adjustment at the biasing.

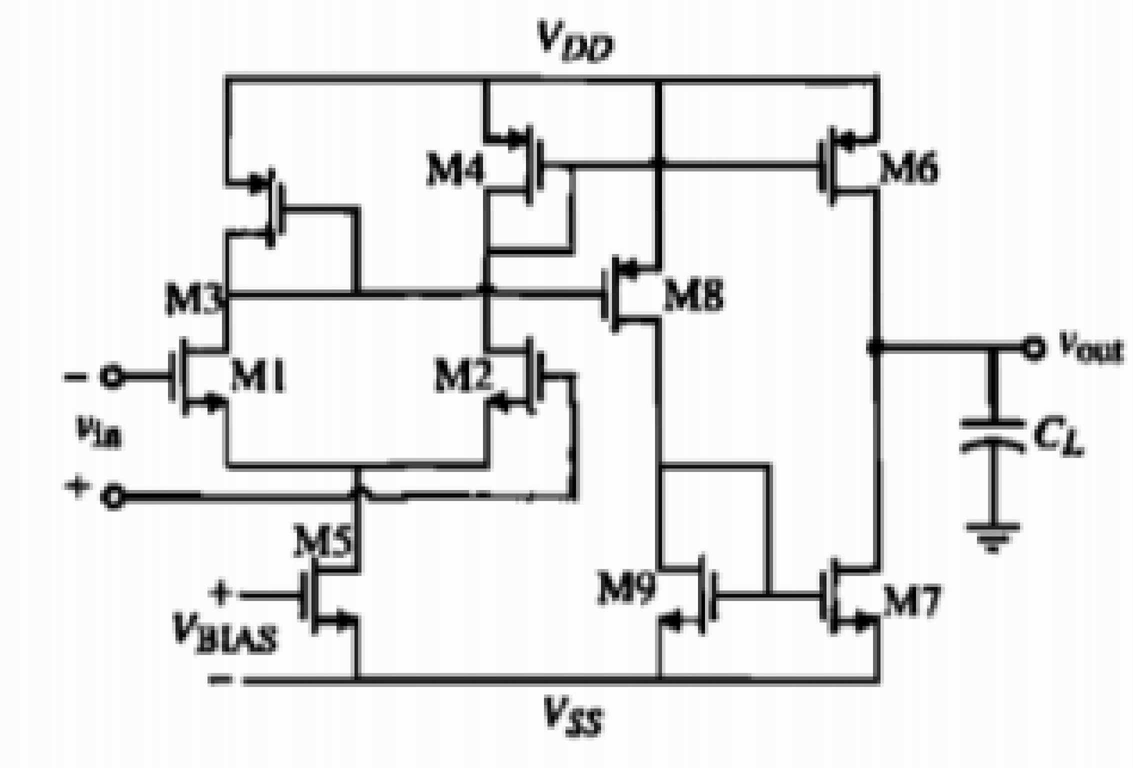


Figure 2.2: The comparator found in P. Allens book [9, p.462]

This is a clamped comparator, some charateristics of it taken from [9, p.462]

The first testbench of the comparator looked like this. Vdd set at 1 V for all simulations, biasing power at -5nA. Both ramps run for 1 millisecond with 1 V gradient.

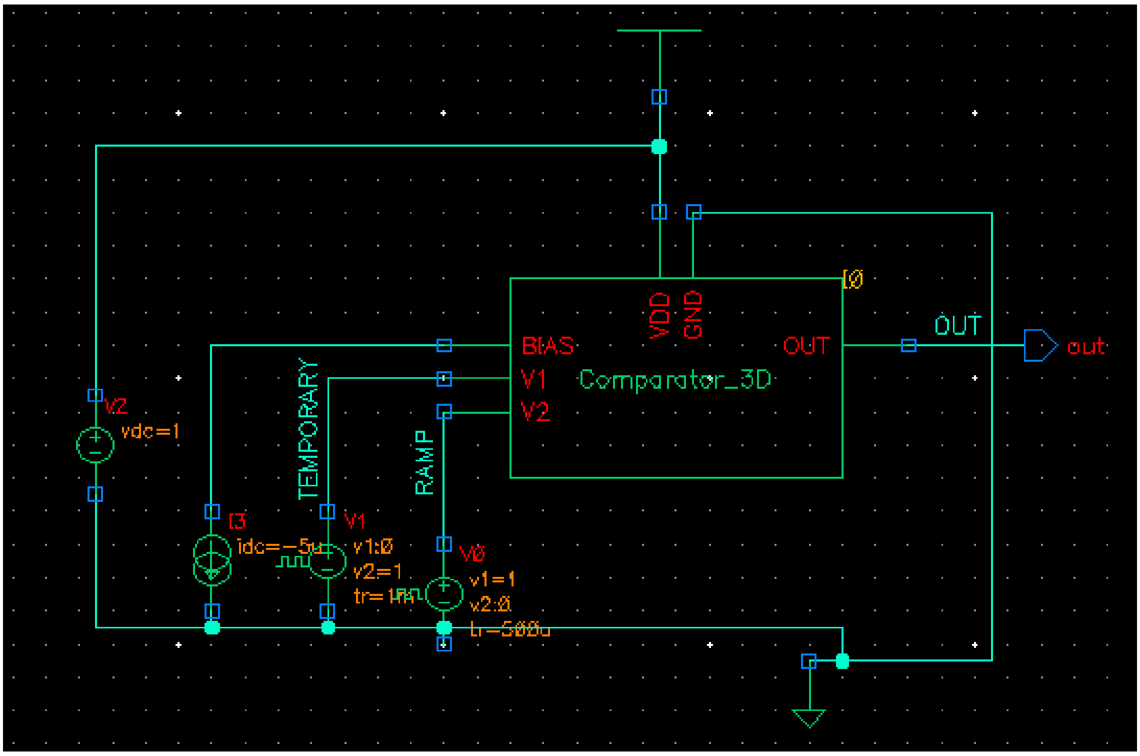


Figure 2.3: Simulating the comparator using two voltage pulse components

In the this test a vpulse element were used as ramp to portrait both the sensor input and the comparison ramp. This was later changed to give a more realistic approach to the input

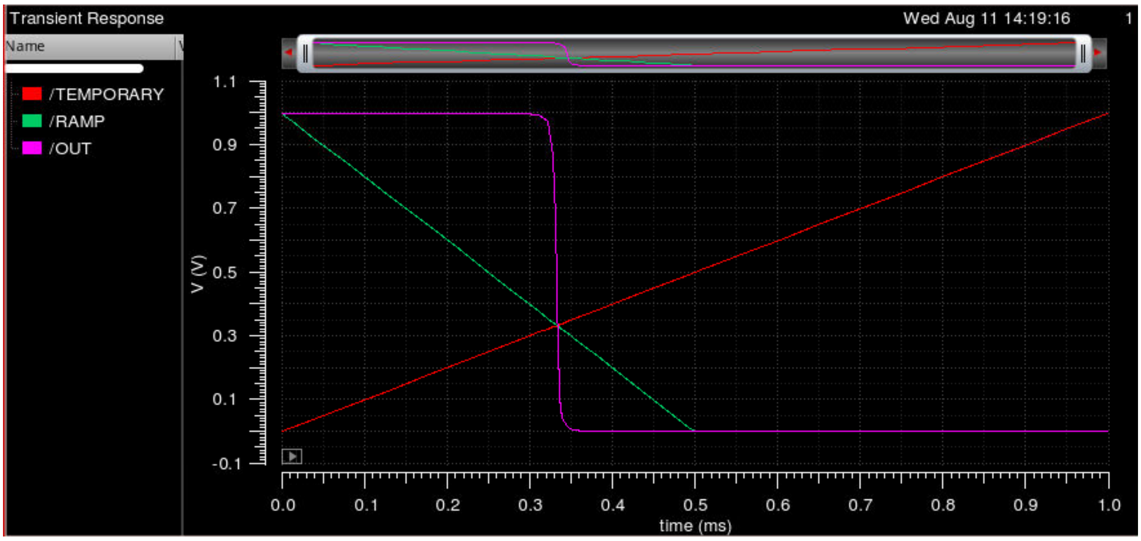


Figure 2.4: Simulating the comparator using ideal voltage ramps

As expected the comparator switches when the two meet, or more accuratly. The value of the comparator output is 0.5 when the value of the input signal received is the same as the comparison ramp. This implicates switching in the counters and essentially stops them.

This was a promising start, the next step was to take the vpulse used for input(falling) out and instead use an idc and a capacitor to simulate the behavior of a photo diode as input.

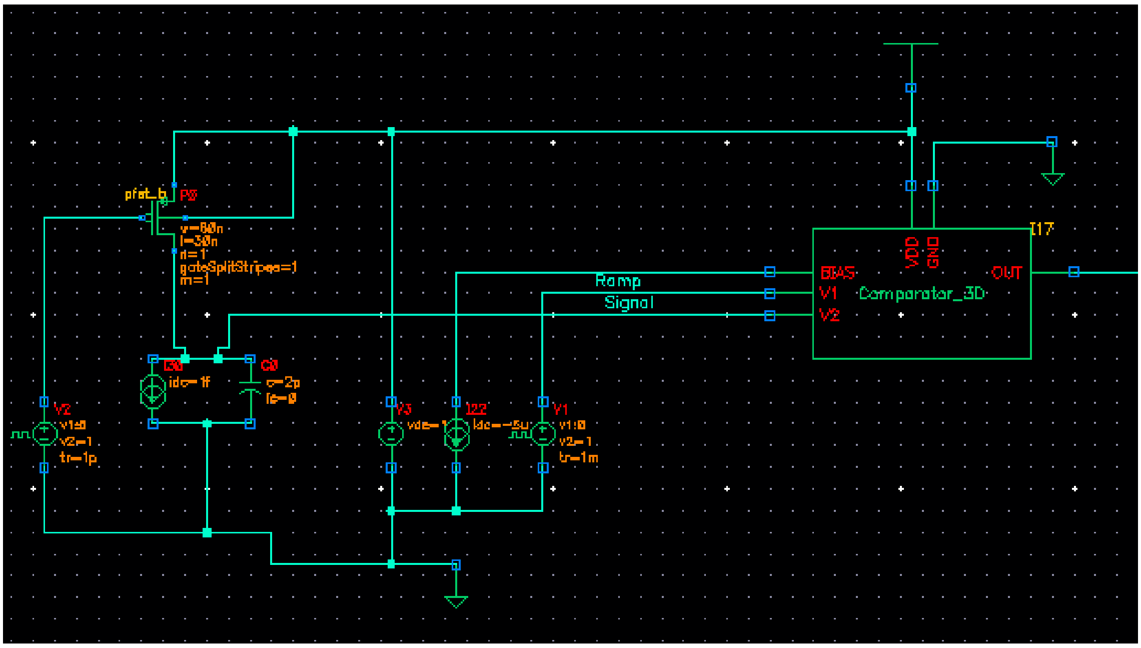


Figure 2.5: Comparator testbench using idc and capacitor to make an input

The input is now depending on the current in the current source which together with the capacitor induces a voltage. This is a way of simulating the behavior of a light sensor.

Proof follows The two forms of the capacitors i-v equation are:

$$i = C\frac{dv}{dt}$$

and

$$v = \frac{1}{C}\int_0^T i\,dt + v0$$

Using number 2, we can find an expression for v when the circuit is operational, t is rising.

In this particular usage I is constant, as its delivered by an Idc, and v0 is zero, (2) simplyfies to

$$v(T) = \frac{i}{C}\int_0^T dt$$

$$v(T) = \frac{i}{C}T$$

The voltage induced is proportional to i/C and such falls like a ramp would.

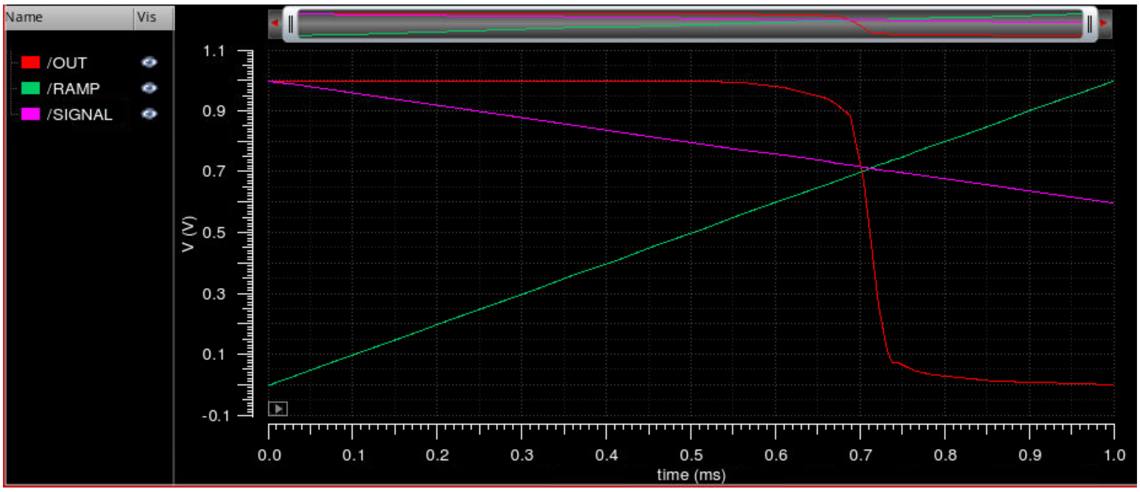


Figure 2.6: Simulating the comparator idc as input

The pfet transistor is used to reset the line by pulling it high when a new input is ready to be recieved after the counting and shifting processes are finished (264,5 clockcycles) controlled by the voltage source V2.

## 2.3 Counter

The counter part of the circuit looks like this in a block diagram.

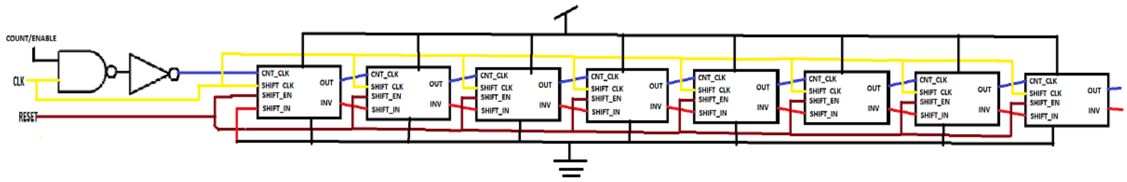


Figure 2.7: Block diagram of 8 counter elements and logic blocks

COUNT/ENALBE is taken from the output of the comparator (fig 2.5). CLK is an external clock signal set to 256 clockcycles in 1ms, giving us a 1kH sampling rate. RESET is also obtained externaly, it is set to change the count blocks from enable to shift/reset every 256,5 clockcycles and holds it for 8 clockcycles to shift(1 cycle per counter element) ENABLE is constructed by sending the comparator output COMP and CLK into a NAND gate. When the comparator is high enable is running, when the comparator switches (goes low) the NAND gate output goes constantly high, this is then inverted and the enable is off.

Having built the flipflops together one of the counting blocks looks like this. CNT_CLK takes the enable signal from the block diagram, SHIFT_IN takes ground and RESET is sent to SHIFT_EN.

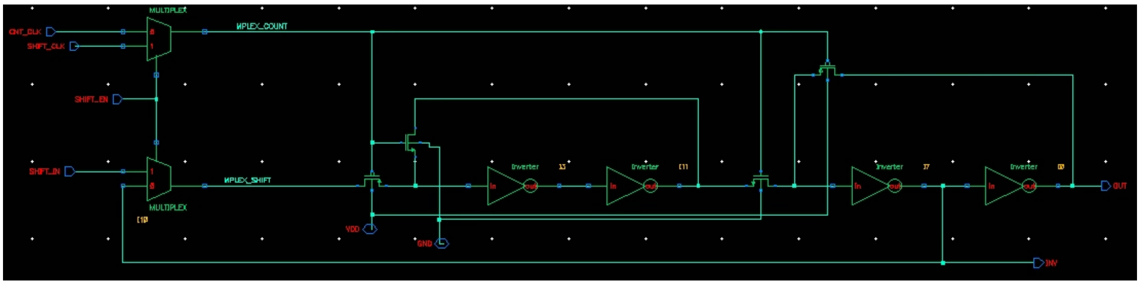


Figure 2.8: Counter element made up of flip-flops

The multiplexers to the left provides the opportunity to switch between counting and shifting.When SHIFT_EN is high the counter is running continuously until the SHIFT_EN signal goes low after 254,5 clockcycles. The extra half clockcycle was necessary to ensure that the counter was finished before resetting. When it goes low the shift register capabilities kicks in and zeroes are shifted in to every cell, after 8 clock cycles the counter is reset conpletely, making sure it will act as we would like when the next signal comes around. Shifting takes a clockcycle per counter cell so 8 clockcycles in this case. After the full 264,5 clockcycles SHIFT_EN goes high again and is ready to start anew.

This functionality is illustrated in section 3.3.1

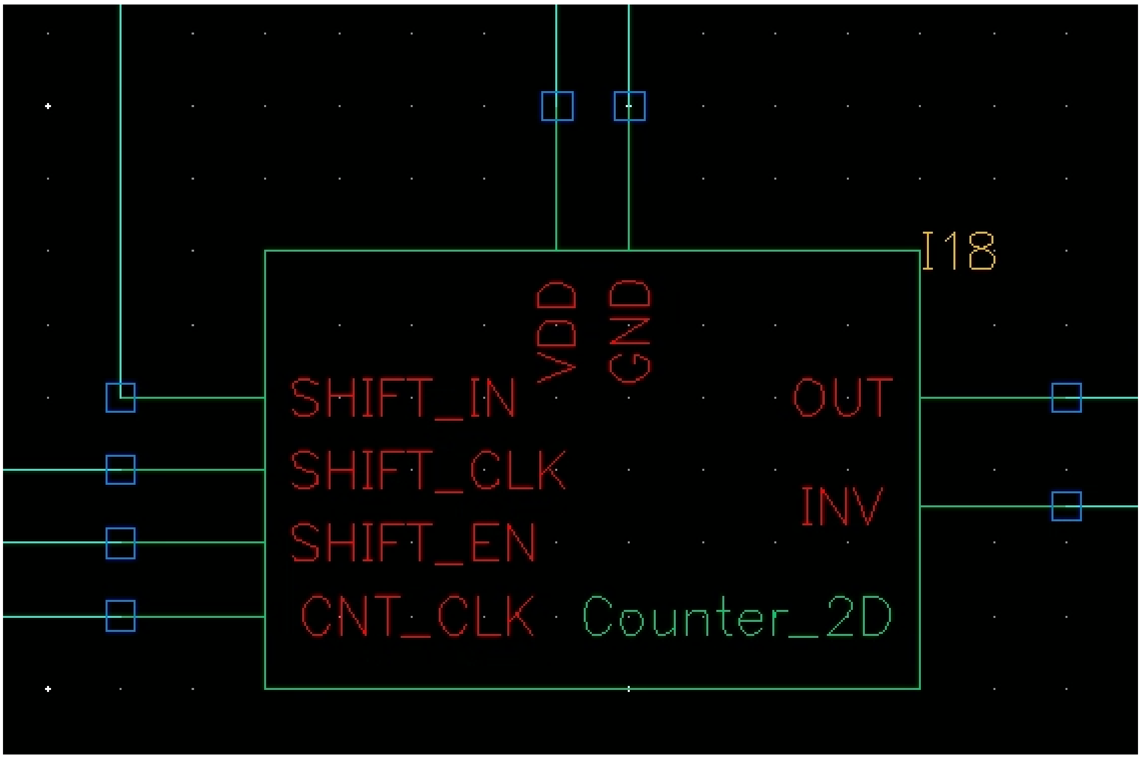


Figure 2.9: A single counter element, the block diagram Fig.2.7 contains 8 of these

CNT_CLK takes enable from the block diagram (Fig.2,7)and SHIFT_IN takes ground in the first counter element. The later ones take OUT and INV respectively. SHIFT_EN takes the RESET signal provided externally to all 8 counter blocks.

Below is simulation results of the maximum possible value of the counter 256. This is obtained when light input is at its lowest intensity, and the induced ramp meets the external at 255 clock cycles.

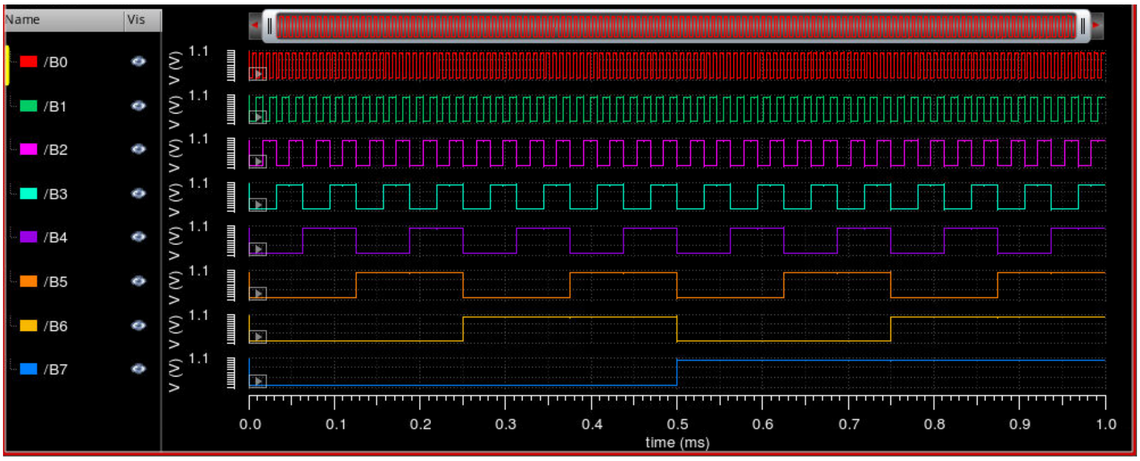


Figure 2.10: Counting to 256 (The plot shows the output of all 8 counter elements simulated)

### 2.3.1 Multiplexer

In this subsection I seek to prove that the multiplexers within the counter elements function correctly.

The multiplexer is consists of four NAND gates connected so they will deliver the following functionality. When select is high (1) and gate 0 is also high (1) then Out = 0. If select is high (1) and gate 1 is high (1) the Out = 1. If all three are high OUT = 1. Other combinations are covered in fig 2.14

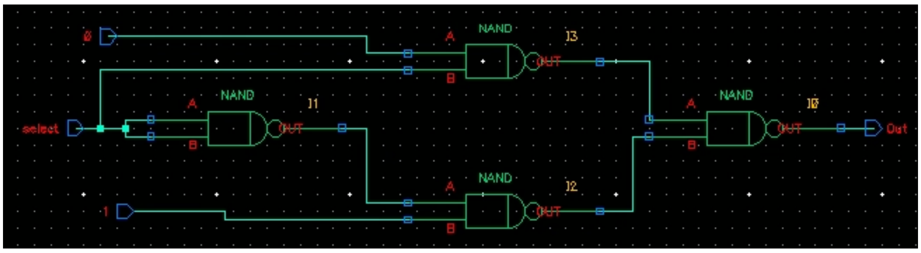


Figure 2.11: Inside multiplexer circuit

The multiplexers are set up as such seen within a single counter block as seen in fig 2.8.

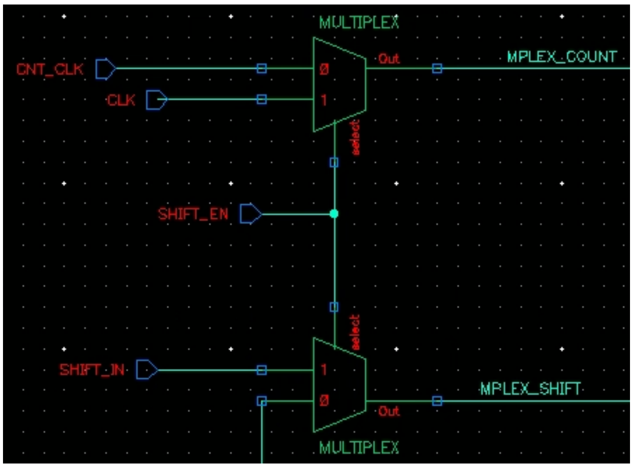


Figure 2.12: Multiplexers which controlls count enable and reset

The upper multiplexer controls count enable and as such cuts of when the comparator switches. Simulating it output in a two counter circuit gives this

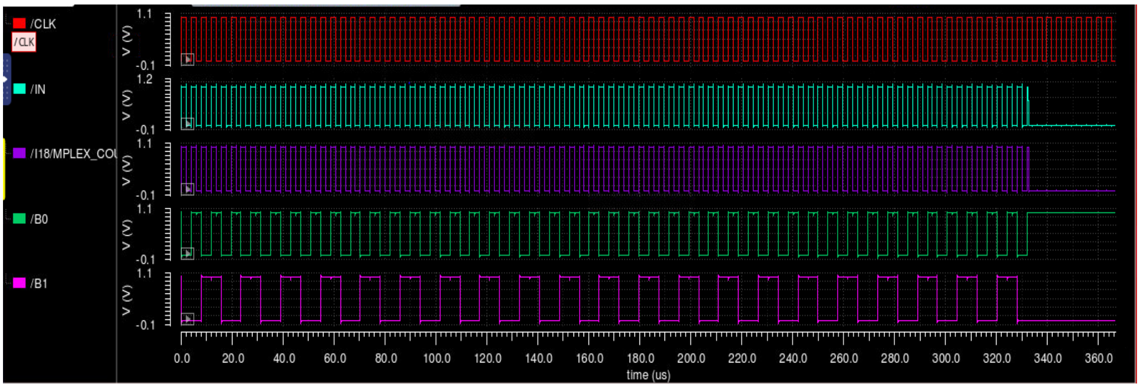
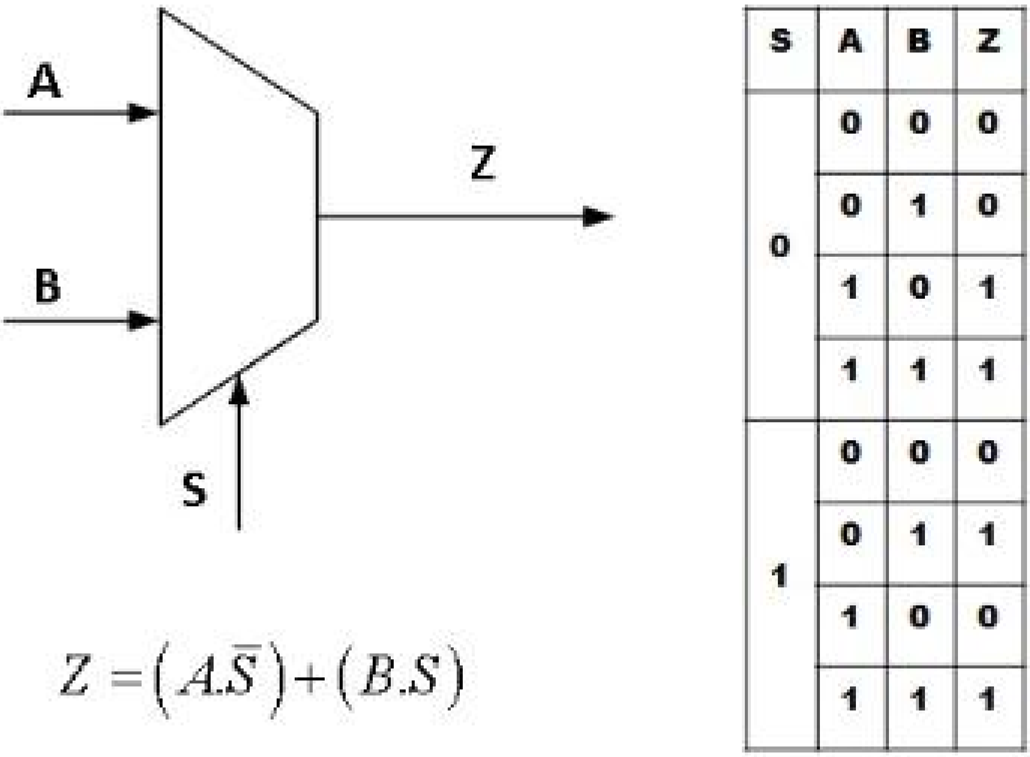


Figure 2.13: Simulation of the multiplexer count enable (Two counters)

Multiplex I9 takes CLK and IN as inputs, in this stage SHIFT_EN is high, as we want the circuit to count.

IN is configured in such a way that it will be as the clock while the comparator is active, and 0 when the comparator switches. As we can read from fig 3.12, when SHIFT_en = 1 (s), CLK are equal while the comparator is high. When the comparator goes low, IN goes low and following the table, the output will also go low for any state of the clock. As we can se from fig 3.11 B0 and B1 aborts their counting at the approriate time.



$$Z = \left(A.\bar{S}\right) + \left(B.S\right)$$

| S | A | B | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
|  | 0 | 1 | 0 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 0 |
|  | 1 | 1 | 1 |

Figure 2.14: Truth tabel for 2-1 multiplex [8]

# Chapter 3

# Results

## 3.1 IO

In finding the IO characteristics of the circuit, i did it by picking out arbitrary values of I for the signal input plotting it against the voltage of the crossing point where the external ramp (RAMP in Fig 3.1) and the integrated input current (SIGNAL in Fig 3.1) cross. The voltage is directly proportional to the time of the crossing and therefore to the counter value which is the ADC output. Figure 3.1 displays an example of the signals involved.

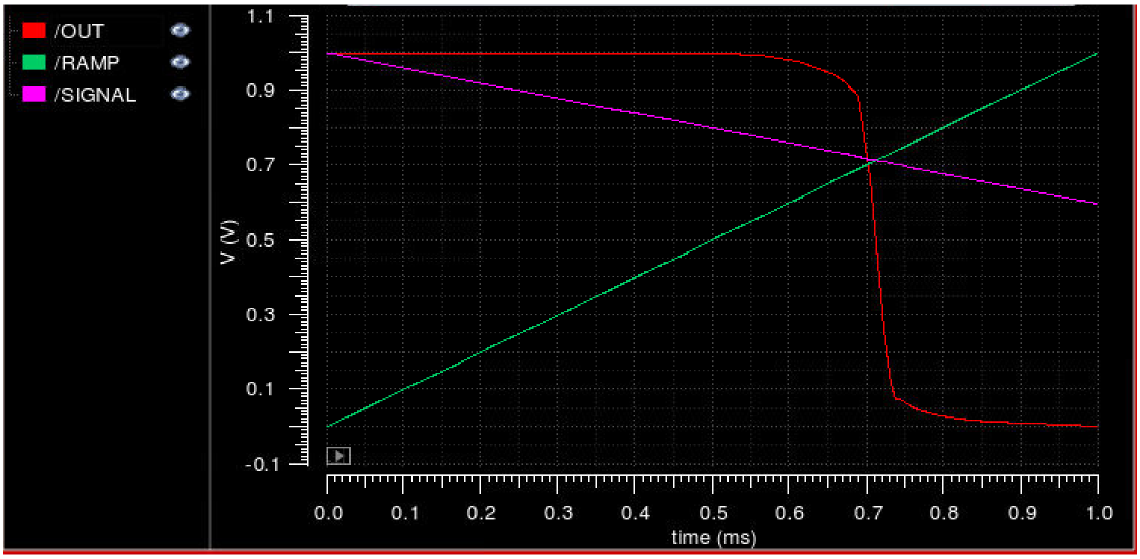
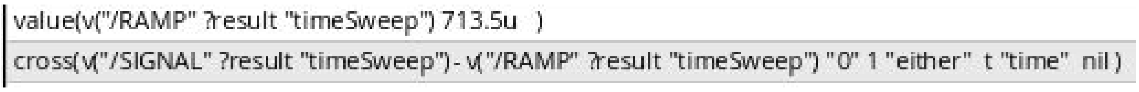


Figure 3.1: Simulation of Fig 2.5 using I30 = 800nA and C0 = 2pF

I then run these two functions through the built in calculator functionality.

The cross function provides the x-value where the two graphs cross. Here this value was x = 713.5u. Using the value function on either SIGNAL or RAMP will return the corresponding y-value to the x-value we give it.

Here it provided y = 713.5m as the value. This is our voltage value for I = 800nA



```
value(v("/RAMP" ?result "timeSweep") 713.5u  )
cross(v("/SIGNAL" ?result "timeSweep")- v("/RAMP" ?result "timeSweep") "0" 1 "either"  t "time"  nil )
```

Figure 3.2: Extracting the voltage value of the crossing point

By repeating this process for many different values of I, concentreting mainly on values which would provide a high voltage or a low voltage I got this graph. X-axis plotted on logarithmic scale. The high and low voltage values are more interesting as the hyperbolic compression leading to a high dynamic range will give small differences there.

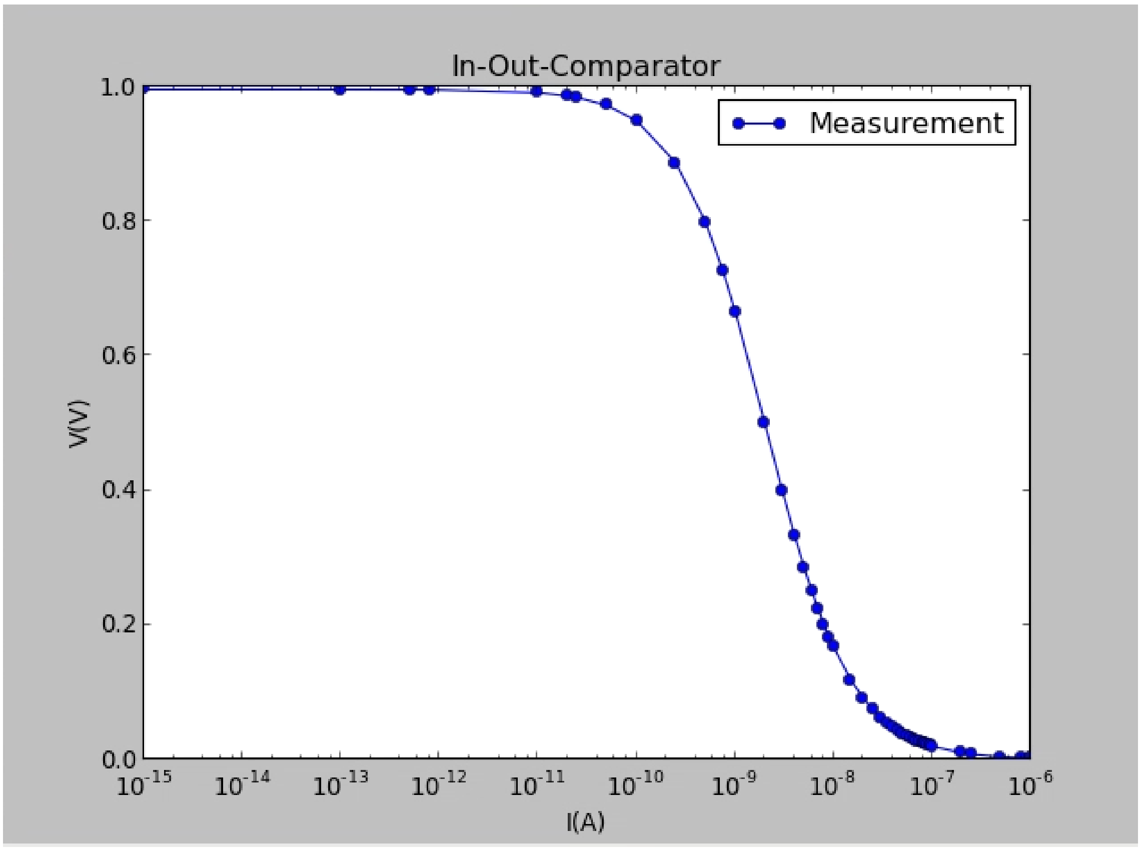


Figure 3.3: IO characteristics of the image sensor

The graph follows what was outlined in [2] (albeit inverted due to the functionality of my circuit) and as such the image sensors gives a hyperbolic compression of the input. Low intensity light will give a high voltage output (high count) and high intensity light gives a low voltage output (low count)

By the IO being hyperbolic the circuit appears capable of securing a high dynamic range to a finished larger scale chip containing several of these.

Using the equation mentioned earlier from [2] for hyperbolic compression

$$h(x) : [0, \infty] \to [0, 1], h(x) = \frac{mx}{mx + 1}$$

I can compare my produced results against theory from [2]. I found m by rewriting the equation to solve for m considering that my circuit provides an inverted result. To find the best value for m, so that the measured value is as close to the theoretical as possible I made a Least Mean Square program (Vedlegg).

Least Mean Square is a way of measuring the average squared difference between the estimated values and the actual values.

To solve it I produced 10 000 blocks of projected y values, and found the smallest Least Mean Square Error, which turned out to be 1.228 *10**9. Which was achieved with m = 501244262.443

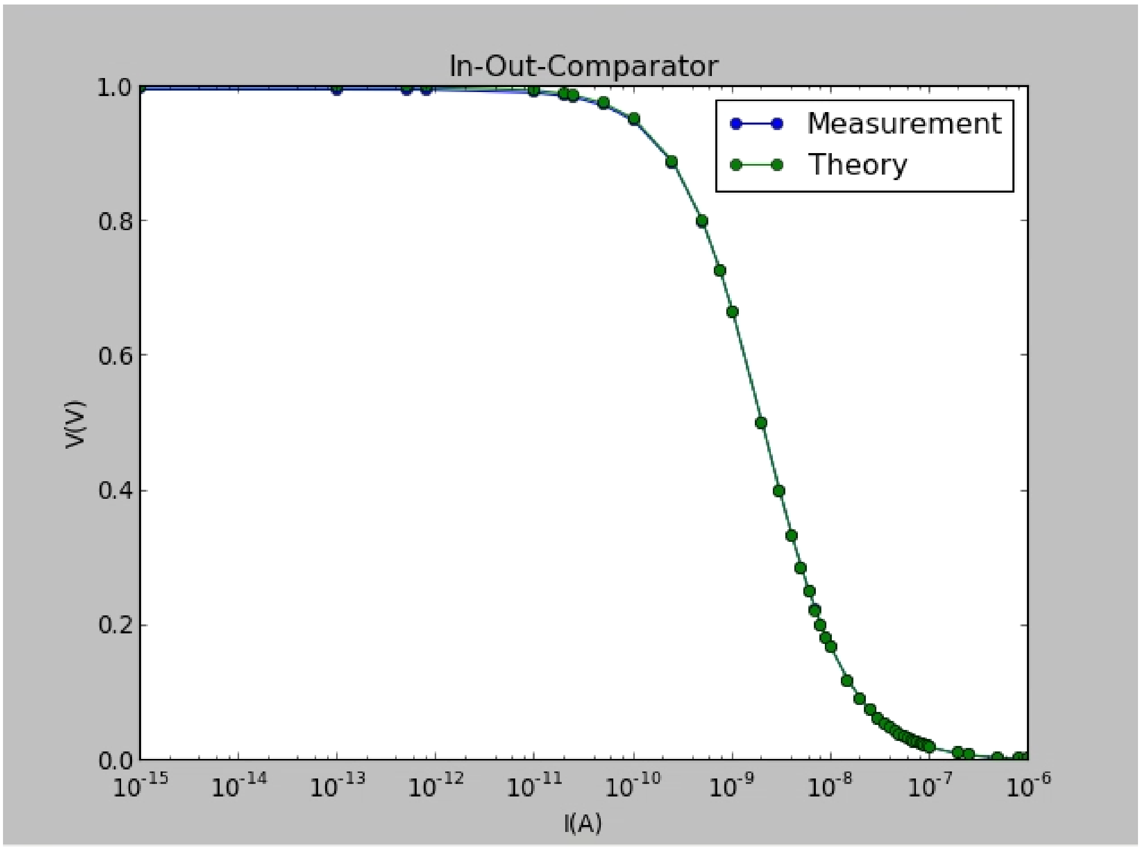


Figure 3.4: IO property of the image sensor comparator

The m used shows that the circuit produced an accurate hyperbolic compression. This shows that the circuitry used is capable of providing a high dynamic range, following theory outlined in [2].

### 3.1.1 Power supply change

Due to how new the technology of 3D CMOS processing is, we do not have any models available for running corners or Monte Carlo simulations for the top tier transistors, instead I attempted to recreate something similar by changing the value of the power supply and the internal ramp. I plotted out 0.8 Volt, 0.9 Volt, 1.1 Volt and 1.2 Volt as well as 1 Volt. This is a way of simulating a 'light' version of a corners simulation and will allow us to quantify and observe possible consequences of noise on the input as this is essentially what is recreated.

In this configuration 0.8 Volt and 0.9 Volt reseambles a slow/slow corner, and the 1.1 Volt and 1.2 Volt a fast/fast corner. The other configurations are hard to emulate in ths fashion, since it is hard to know which transistors go fast and which go slow in fast/slow and slow/fast cases separatly.

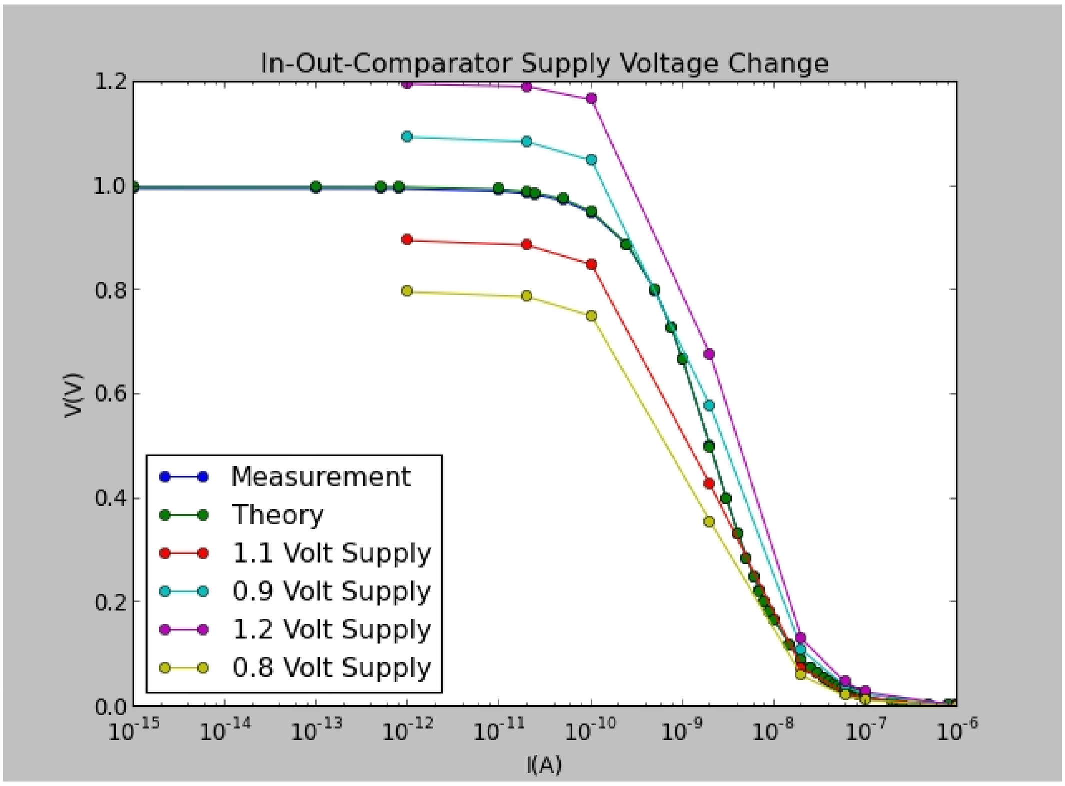


Figure 3.5: Change in Supply Voltage

As seen a higher supply voltage gives higher voltages on the output for the same the same current input as a lower supply voltage would. The drop is steeper at values in the 100 pico to 2 nano ampere input range, influencing the sensitivity of the sensor. This is as expected and is explored further later i the paper by looking at the time shift on the output of the comparator.

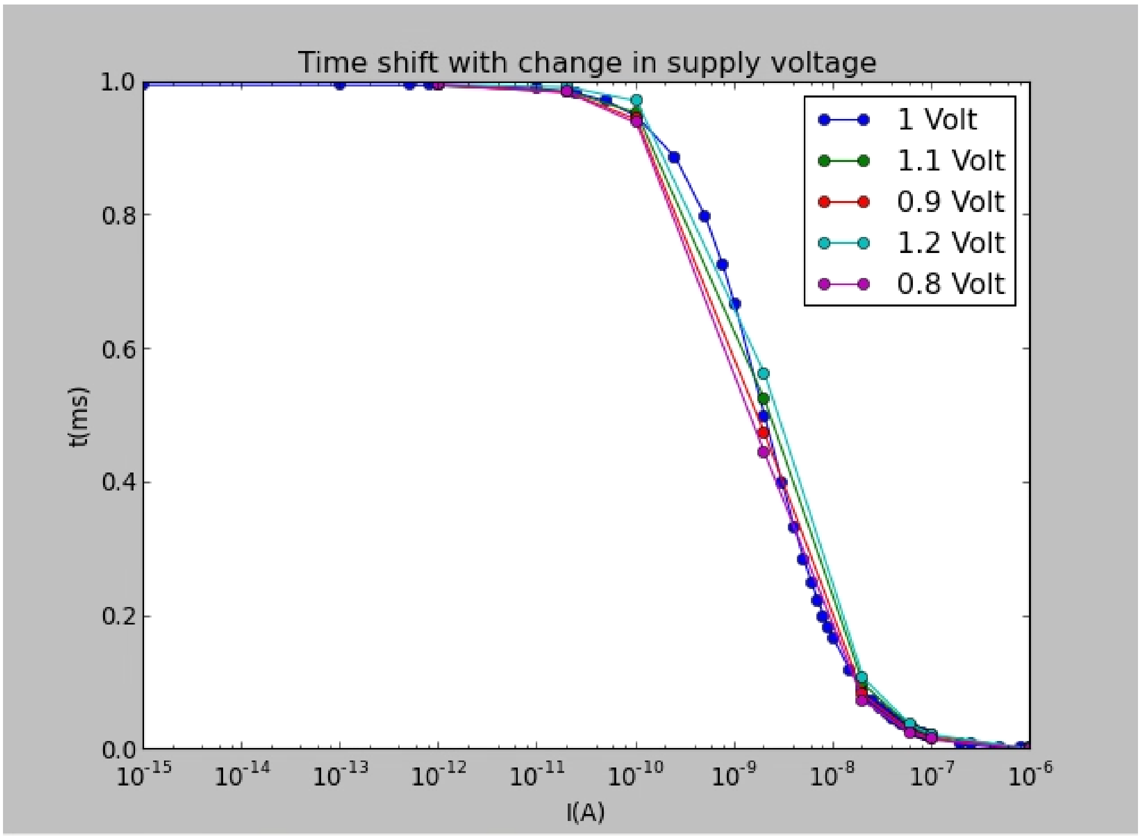


Figure 3.6: Time shift at different voltages

Looking at the time signatures of different supply voltages we observe that at higher voltages everything happens slower, because the value is simply larger and it takes longer for the internal ramp and the photo current to meet. The time signatures are just the voltage values scaled to match the 1 Volt. As we can see the graphs are still not exact copies, even after beeing scaled, this has implifications on the predicted performance of the sensor. Results of this covered further below.

This can also be displayed in a Least Significant Bit plot, since time and number of bits are directly correlated since a bit is just the time it takes to run a full operation over the number of bits used. 1 bit portrayed in time is 1ms/256 = 3.90625*10^-6.

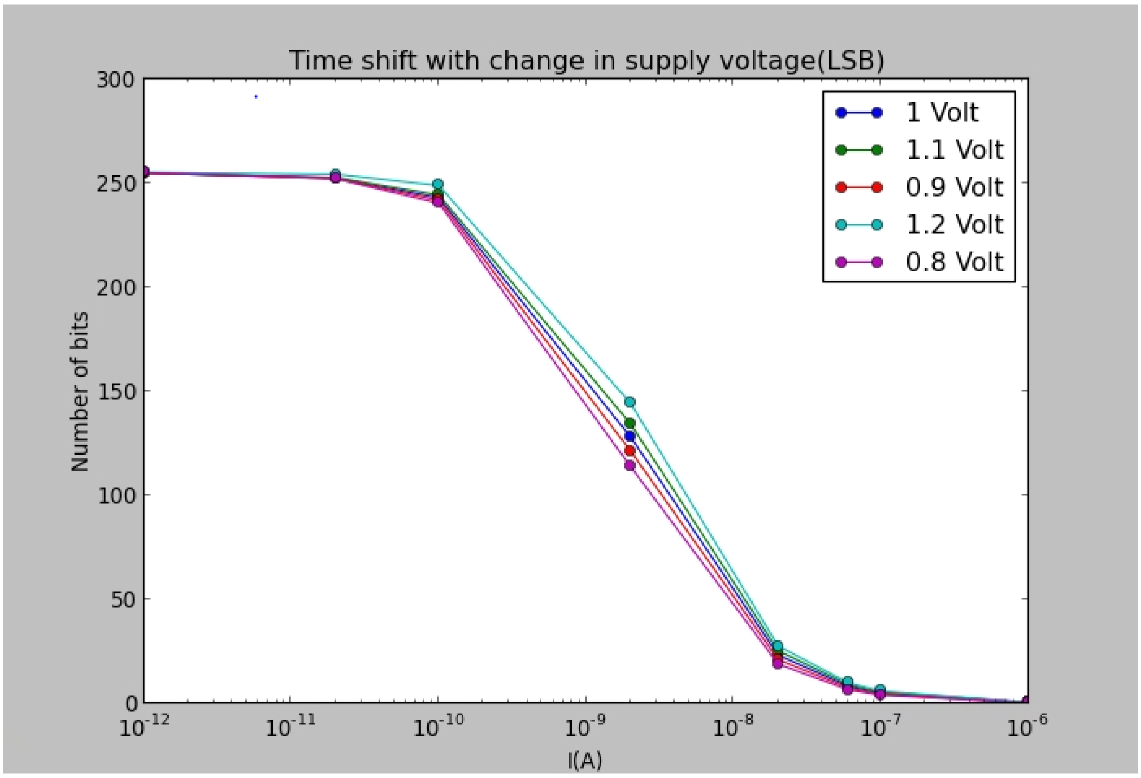


Figure 3.7: Time shift at different voltages portrayed as number of bits

The number of bits needed to complete full count is highest at small current inputs as these give the highest voltage output and such longer time and more bits, this is consistent with fig 3.4 As seen in Fig 3.7, the offset around the points with highest sensitivety are clearly off by a number of bits. Shown in Fig. 3.9.

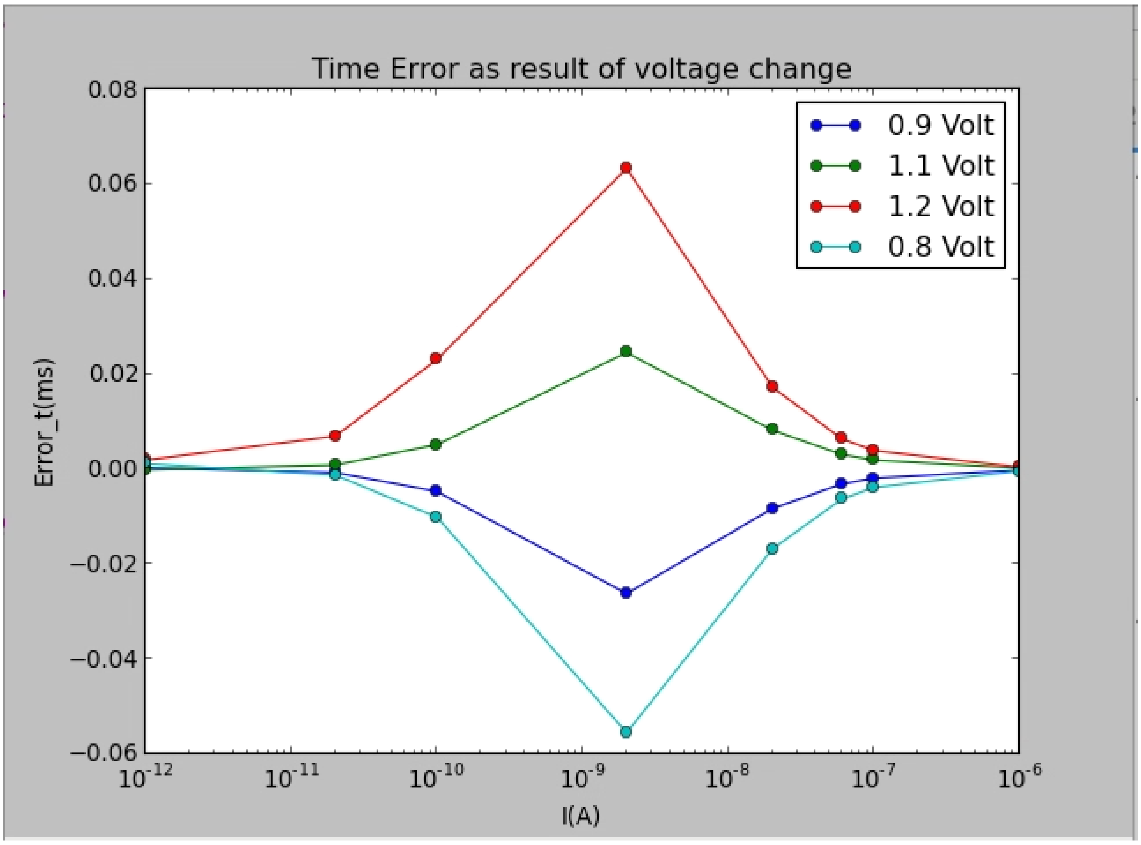


Figure 3.8: Error in timing with change in supply voltage

As we observe after the scaling operation (Fig 3.6) there are still errors compared to the basic 1 Volt operation. This error is interesting as we can use it to to find an connection to the sensitivity of the sensor at different voltages.

The error is larger at the middeling values of I rather than at the two extremes. At 1 pico and 1 micro ampere it is close to 0, while in the 50 nano range it goes as high as 0.063 ms. This is because the sensitivety of the sensor is higher there. This makes sense as sensitivity in this application is defined as how much the sensor's output changes when the input quantity being measured changes. In fig 3.6 we see that a current change at minimum or maximum value, will hardly change how the y-axis corresponds. However in the middle a small current change can correspond to a much larger change, the error image corroberates this. It is also worth noting that the further we get from 1 Volt either side the error increases.

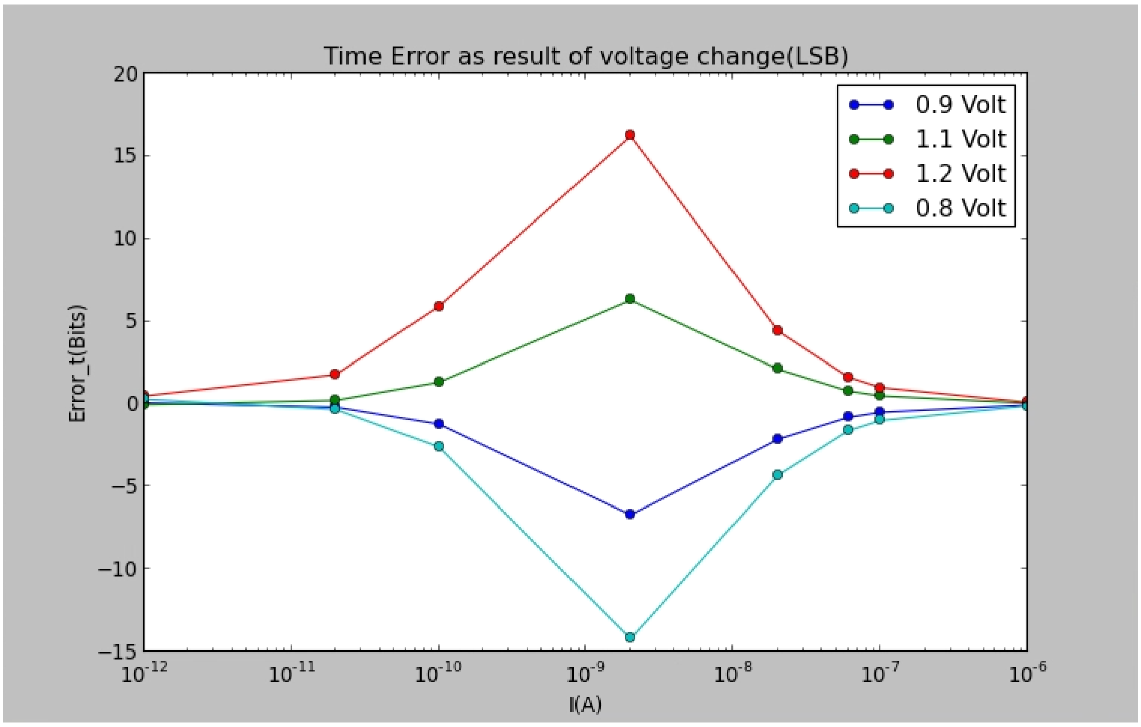


Figure 3.9: Error in timing with change in supply voltage given in number of LSB

At 10% supply voltage the error is 6.2 LSB at peak, at 10% supply voltage decrease the error is 6.7 LSB at peak. 20% supply voltage increase gives 16.2 LSB error at middeling input values and 20% supply voltage decrease 14.2 LSB error at peak compared to the 1 V measurement.

Now using equation (11) from Philipps paper [2], and looking at the step size corresponding to the LSB of the output, we can say even more regarding the change in contrast over time for the circuit and implication of the error obtained through supply voltage change.

$$\Delta^{hyp} x \approx \frac{2^{-N}(mx+1)^2}{m}$$

Plotting this using the m obtained through the Least Mean Square program (m = 501244262.443), the x obtained originaly and N = 9 produces this image (Fig 3.10).

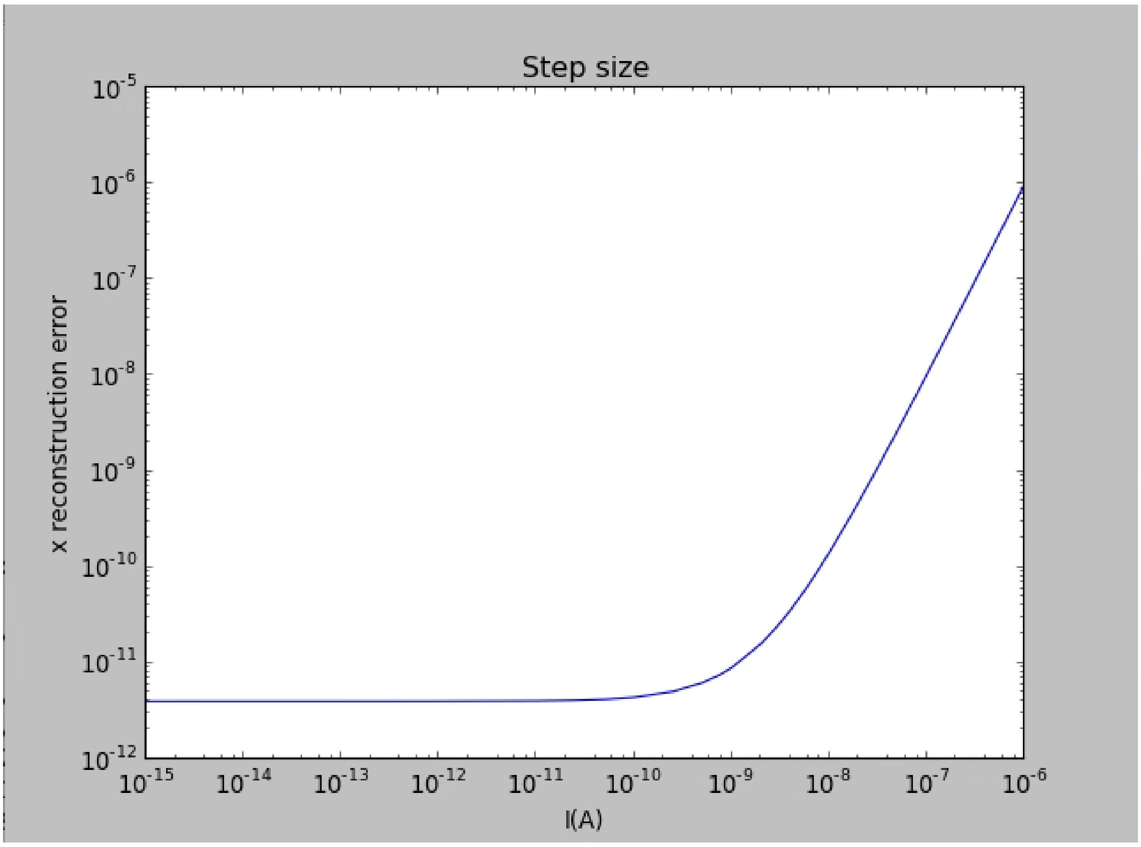


Figure 3.10: Step size plot of LSB to find contrast

Fig 3.10 displays the step-size given in LSB and is further used to determine the change in contrast at different inputs. Now using this result and dividing it by the original x, we get an idea of how the contrast changes at different voltage inputs.

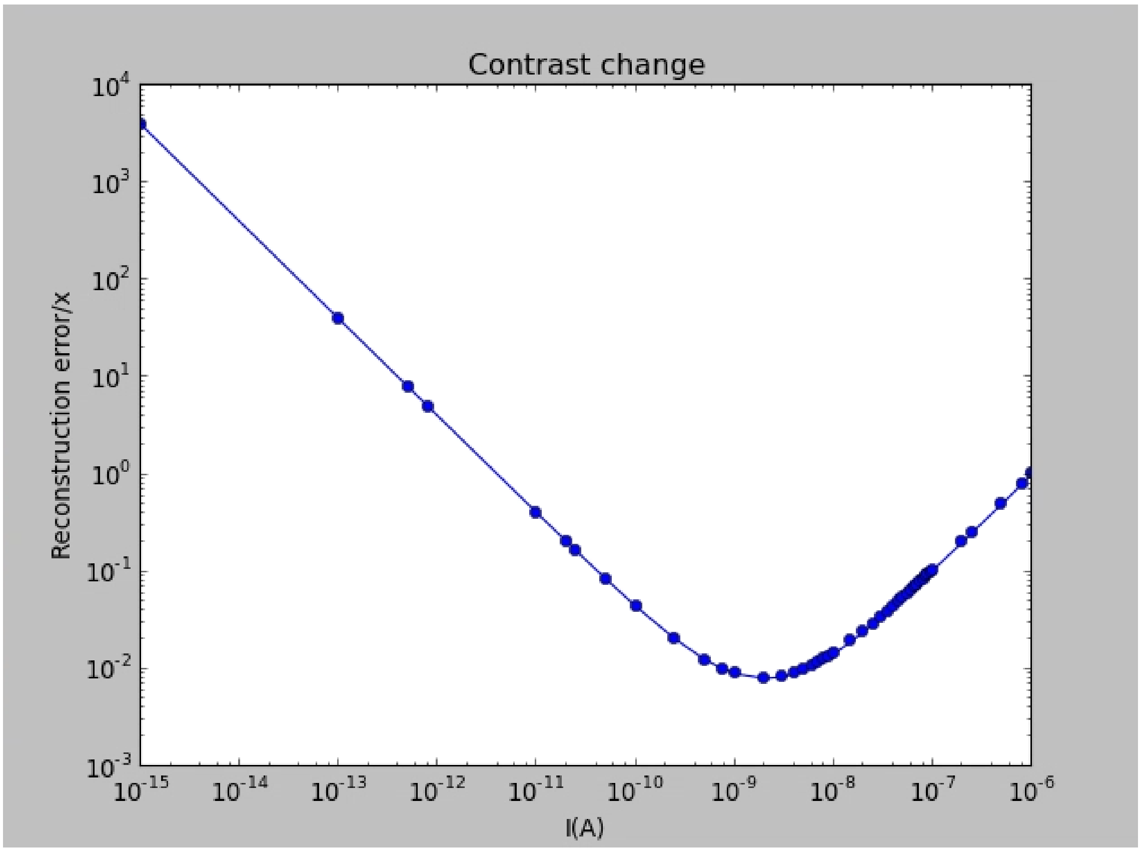


Figure 3.11: Contrast change in supply voltage/ digitization error

Contrast is a measure of relative difference in value. How much does the input value need to change in order for us to notice a difference on the output (LSB switching). As we can se from Fig 3.11 at voltages in the pico amperer range and lower, the change in input needed to affect the output is more than 1000% percent. This is the darkest area and as such we would expect that a large change in luminosity is needed to change the output. As the input increases, the output is affected more and more, until the input area of maximum contrast sensitivety at 50 nA where the LSB corresponds to a relative change of below 1%.

This is where the smallest change on the input will affect the output. If we compare it to fig 3.8 we observe that it coincides with where the largest timing error occured. This is not a coincidence, this is where the sensitivity is highest and such this is where there ouput is subject to most change. At higher voltage inputs it again takes more to change the output LSB as we are nearing the lighter areas.

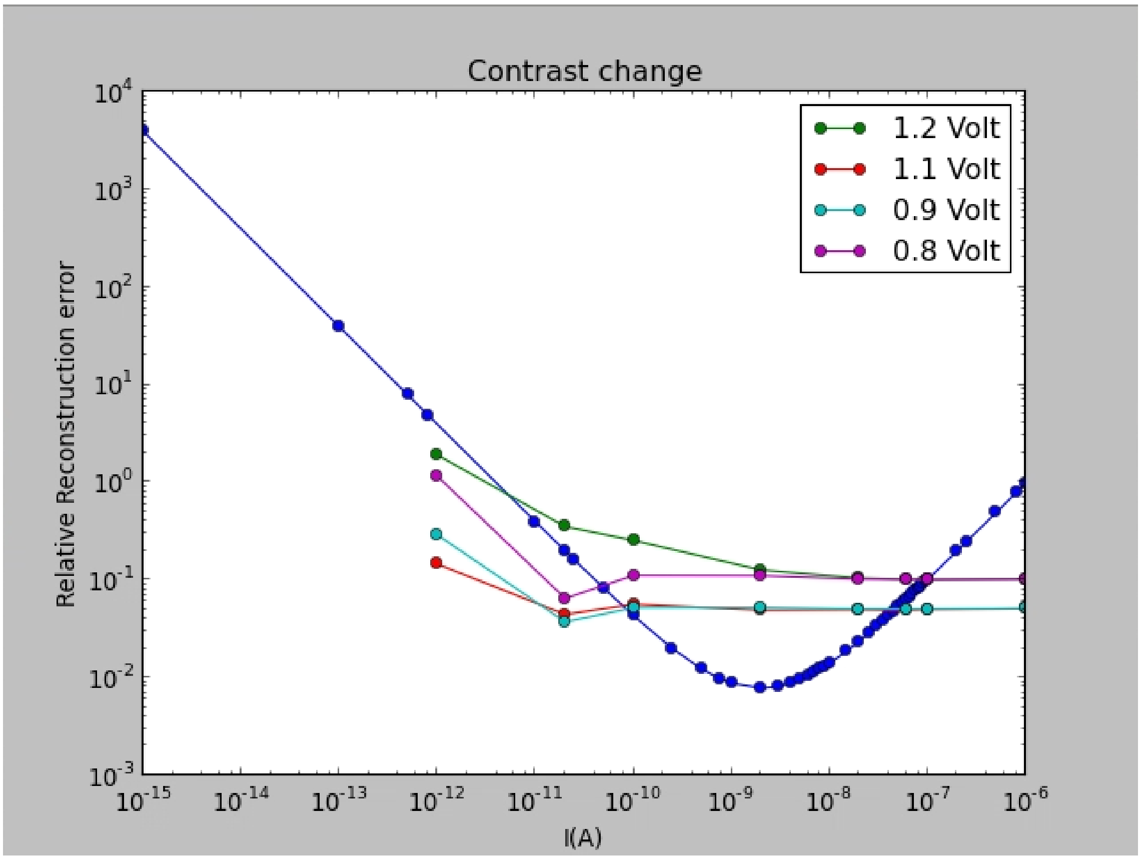


Figure 3.12: Contrast change and error*contrast displayed in LSB

Plotting the timing error (fig 3.9) given in LSB multiplied by the contrast found through the step-size function we are able to make some observations. The time offset introduced by changing the supply voltage affects the contrast on the output of the circuit directly and greatly. I introduced the voltage changes as a static change, to simulate possible noise problems which may come up. As seen in Fig 3.12 the error from the noise introduced is larger than the digitization error for the central values of the input, essentially introducing a contrast-noise floor at these values, at the extremeties of the input range the digitization error is larger than the noise induced error. As seen in Fig 3.9 the error is highest were the sensitivety is highest. The sensitivety is reduced in the central areas due to the noise introduced by the supply voltage changing.

# Part II

# Conclusion

# Chapter 4

# Discussion

## 4.1 What I found out

As seen in sections 2.3 and 2.4 the circuit functions as I wanted it to. It is able to convert an analog photoelectric input to an 8 bit digital value through a comparator and a block of counter elements. The circuits function is consistent with the theory found in [2] with a mean square voltage error of the crossing point of the ramps where the full range is 1V of 1.228*10$^{-9}$ V after parameter fitting theory and the achieved data. Which means that the mean square error before digitization in LSB is insignificant.

The PDK did not allow Monte Carlo simulation, corner analysis or transient noise. However to verify the robustness of the circuit i analysed in influence of a change in power supply on the circuit. Plus/minus 10% variation resulted in a loss of 7 and 6 LSB respectively in the central area input range where the sensitivity is highest and less then 1 LSB at the lower and higher range.

Actually when considering the noise expressed in contrast, it looks like noise/offset on Vdd directly leads to a constant error in contrast over large parts of the input range, effectively introducing a contrast-noise floor. Increasing the noise on Vdd causes that noise to become dominant in the centre of the input range where digitization error in terms of contrast is minimal. The digitization noise remains dominant on the outer extremes of the input range (fig 3.12). Increasing the noise on Vdd gradually enlarges the area affected by the contrast-noise floor.

However due to this thesis only providing simulation of schematics it is hard to say, if this is all applicape to a real life circuit.

## 4.2 Future Work

Future work on the project would include, producing a layout, production of circuit (currently unavailable for combination of top tier 65nm transistors and bottom tier 28nm) and performing complete Monte Carlo and corner simulation as these are also currently unavailable

This would all be interesting, as it would be possible to determine the fill factor, which would be close to 100%. Further out of reach at present is technology to combine three tiers to allow a back side illuminated photo diode array This would also have brought the circuit closer to a real world application.

# Bibliography

[1] Basics and overview of flip-flops Electronics for U
https://www.electronicsforu.com/technology-trends/learn-electronics/flip-flop-rs-jk-t-d

[2] Phillip Dominic Häfliger, Mikkel Mikkelsen
In-Pixel ADC Concept with HDR Hyperbolic Compression and CDS
https://epapers.org/mwscas2021/ESR/paper_details.php?PHPSESSID=jmnl5t3dfn96v5van6o5loqmt2&paper_id=2053

[3] Mikkel Mikkelsen
Pixel-Parallel Analog til digital konvertering i CMOS bildsensor https://www.duo.uio.no/handle/10852/70744

[4] Wavelenght electronics: Photodiode basics https://www.teamwavelength.com/photodiode-basics/

[5] Adel S.Sedra & Kenneth C. Smith (2016), *Microelectronic circuits*(International seventh edition).Oxford University Press

[6] Wikipedia CMOS: Wikipedia the free encyclopedia
https://en.wikipedia.org/wiki/CMOS

[7] Wikipedia
High-dynamic-range imaging: Wikipedia the free encyclopedia
https://en.wikipedia.org/wiki/High-dynamic-range_imaging

[8] Virtual lab: 4x1 Multiplexer https://vlsi-iitg.vlabs.ac.in/Multiplexer_theory.html

[9] Phillip E Allen & Douglas R Holberg *CMOS Analog Circuit design*
http://digitus.itk.ppke.hu/~karzo/VLSI_books/Allen,%20Holberg%20-%20CMOS%20Analog%20Circuit%20Design%20second%20edition.pdf

[10] Circuitdigest: Asynchronous counter https://circuitdigest.com/tutorial/asynchronous-counter

[11] Wikipedia
Counter (digital): wikipedia the free encyclopedia
https://en.wikipedia.org/wiki/Counter_(digital)

[12] Wikipedia
Analog-to-digital-converter : Wikipedia the free encyclopedia https://en.wikipedia.org/wiki/Analog-to-digital_converter

[13] 3D-MUSE Key concepts
https://www.3dmuse.eu/outline/index.html#section_architectures