NTNU – Trondheim
Norwegian University of
Science and Technology

TFE4152: Design of integrated circuits
Term Project

Digital Camera

Group 30:

Cristian Gil Morales
Hector Mercado Valls

November 21, 2018

Abstract

It is our job to make the solution a detailed presentation of the problem

The present report gives a solution of how to make a digital camera work properly by studying part of its analog and digital design.

This approach consists of the parametrization in the analog circuit and a HDL code, simulating the control logic behind the analog circuit.

The academic aim of this project is to learn how both analog and digital electronics work together in order to provide a fully functional system.

Contents

List of Figures

List of Tables

1 Introduction

In the light of absent objectives in the specification of the project, some assumptions need to be done in order to define a problem:

- A digital camera is generally used for the next purpose:
  Capture a visual instant of the analog world in order to have a digital persistent resource called ***image***
- The general quality expectations from an image took from a digital camera are the following:
  The ***image*** *contains an accurate visual representation of the analog world at the instant in which the user pressed the "Take_ a_ picture" button*

1.1 Description of the problem

Therefore the main project problem can be defined as satisfying this quality standard. However, how can this project, given the technical specifications, cope with this problem?

1.1.1 Analog part

The main focus of the project description is limited to the link between the analog sensor and the ADC in Figure 1.1, and the logic required by the digital camera. It is obvious from Figure 1.2 that the choice of the transistor sizes, sampling capacitances and exposure time is going to influence the analog voltage signal that ADC has convert to binary through quantization. Thus the main problem for the analog part can be more strictly defined as:

Every pixel of an image has to be linear with respect to the incident light of the respective pixel photodiode at the shooting instant, in order to obtain an accurate, proportional, undistorted representation of the analog world.

From this observation it is possible to subdivide the main problem into 3 subproblems:

1. The image is composed of pixels. Each pixel represents the information of brightness that its respective sampling capacitor captured through its photodiode during a exposure time. In order to acquire a consistent representation of the analog world it is defined the following **constraints that must be satisfied**:
   a) For every photodiode, if photodiode A is exposed to the same light conditions as photodiode B, both photodiodes should generate the same current intensity.

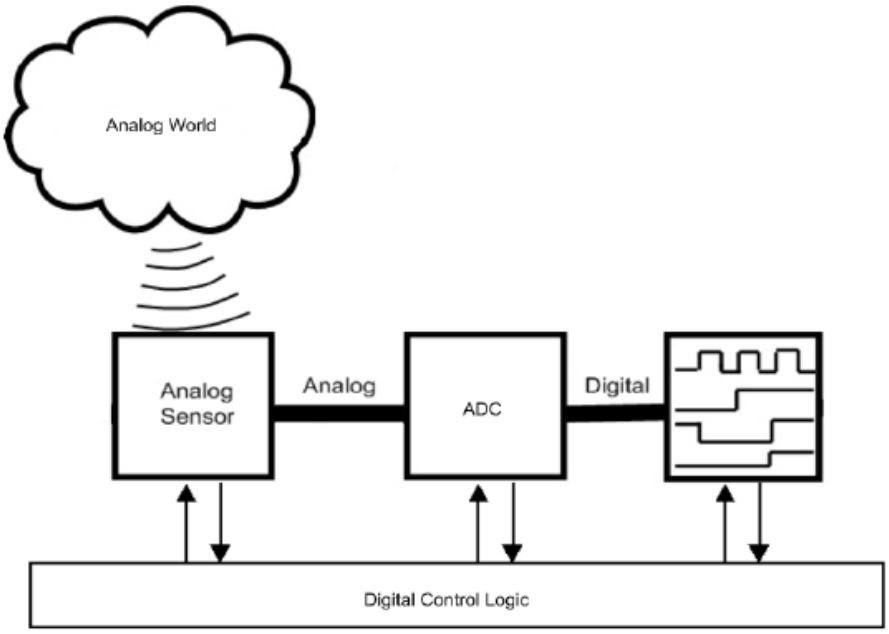


Figure 1.1: Simplified view of a digital camera

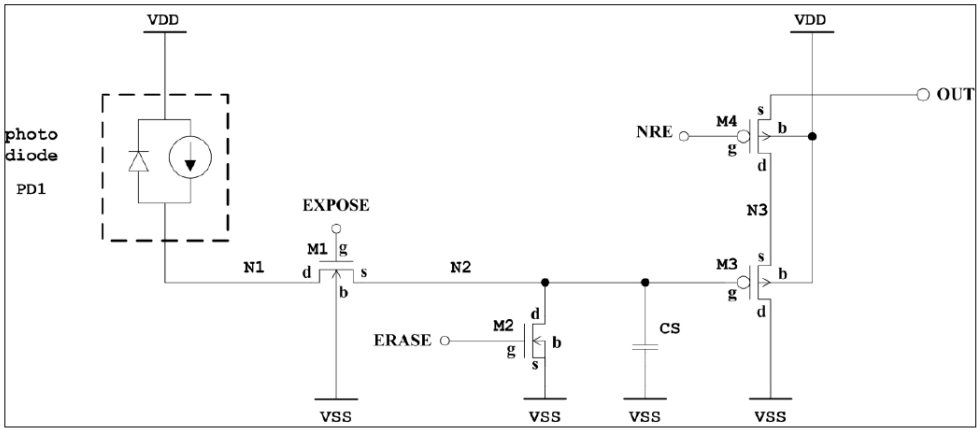


Figure 1.2: Pixel circuit diagram

b) The quantifying discards information by approaching a continuous variable to a discrete value. The input voltage of the ADC should be a monotonously increasing function of the brightness at every pixel so that discretization errors are consistent.

2. Due to the exposure time, a delay is introduced between the push of the button and the creation of the image. As the user wants an image from the instant that the button is pressed, one should **minimize the exposure time**.

3. If the exposure time is minimized too much during poor light conditions, one may obtain an **underexposed** image, whereas an excess of exposure in good light conditions may lead to **overexposure**.

1.1.2 Digital part

Inside a digital camera there is an analog part to interact with the real world (and it will always be unavoidable as the real world is analog), but, as its name already suggests, the rest of the camera (the biggest part) is digital. This circuitry is crucial in order to ensure all analog data is taken properly synchronized and transformed into the digital world.

After this digitization, the data is processed to create the picture one can enjoy later, apply filters, etc. (in any case, that would be already out of the scope of this project).

The following picture shows the requested digital inputs and outputs, and how are they supposed to interact each other:

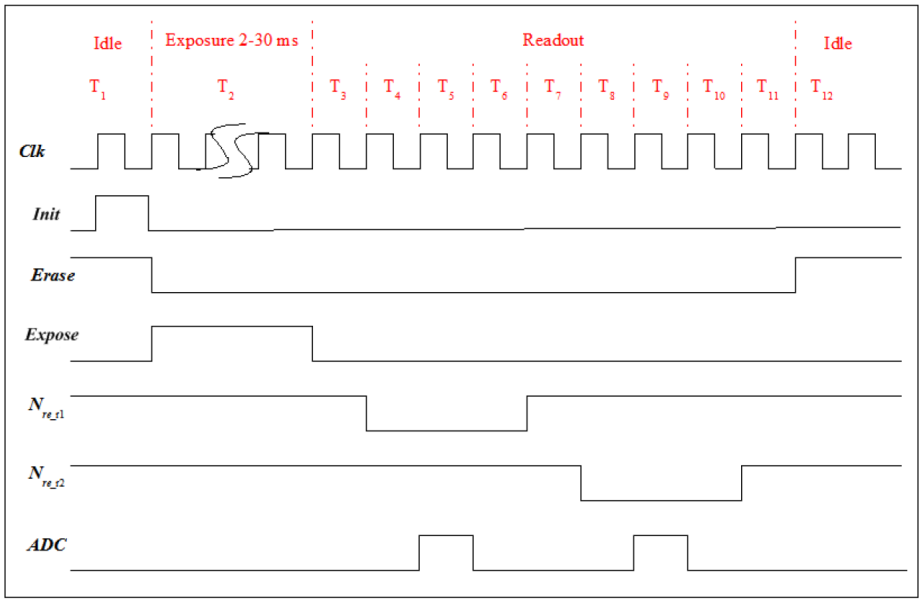


Figure 1.3: Timing diagram of the digital system

Overview of the inputs and outputs:
INPUTS:

- Clk → Clock signal, which allows the synchronization of the whole system. The requested frequency is 1kHz.
- Init → Initialization signal. A 1ms input signal (out of scope), that starts the process to take a picture. It is a physical button in the digital camera.
- Exp_increase → It increases the exposure time of light up to 30 ms, so that the image gets clearer. It is a physical button in the digital camera.
- Exp_decrease → It decreases the exposure time of light up to 2 ms, so that the image gets darker. It is a physical button in the digital camera.

OUTPUTS:

- Erase → It discharges the capacitor Cs, clearing the analog system for taking the next picture. For security reasons, this signal is permanently 1 during stand-by.
- Expose → It allows the photodiode to charge the capacitor Cs, storing on it the new data to be processed. This signal must be 1 only during the light reading process, which can last from 2ms to 30ms.
- NRE_{R1} → It activates the first pixel of each column at a time (pixels 1.1 and 1.2) so as to send the data stored in capacitor Cs to the ADC converter. Activation time is shown in Figure 1.3, with a specific duration.

- NRE_{R2} → It activates the second pixel of each column at a time (pixels 2.1 and 2.2) so as to send the data stored in capacitor Cs to the ADC converter. Activation time is shown in Figure 1.3, with a specific duration.
- ADC → It enables the 2 ADCs to process the analog signal received from both NREs at a time, so its activation time is during both NREs activation times too (also shown in Figure 1.3).

Figure 1.3 also divides the whole functionality of the digital system into 3 different STATES:

- IDLE → The system is in stand-by, waiting for the user to take the next picture. In this state, the output "Erase" is permanently 1 to ensure that the capacitor Cs is completely discharged. At the same time, inputs "Exp_increase" and "Exp_decrease" are enabled to change the exposure time, as well as input "Init" to take the new picture.
- EXPOSURE → The process of a new picture has already started and this state toggles outputs "Erase" and "Exposure" during a time from 2ms to 30ms (defined in IDLE), charging the capacitor Cs (less or more time respectively) through the Photodiode installed in the camera. The rest of I/O are disabled.
- READOUT → With the capacitor already charged, the output "Exposure" is also disconnected to ensure this voltage is not lost, and the outputs "NRE_{R1}", "NRE_{R2}" and "ADC" are activated and deactivated in a particular order (as shown in Figure 1.3) to do two sendings of two pixels at a time (a total of 4 pixels) so that the ADC digitizes this information.

1.2 Specifications

The requested solution must fulfill the following requirements:

- Analog design with 180nm transistor technology
- For all transistors, sizes must be between 0.7μm<L<2μm and 2μm<W<10μm
- For all sampling capacitances, $C_S \leq 3pF$
- Column capacitances, $C_{C1} = C_{C2} = 3pF$
- Supply voltage, $V_{DD} = 1.8V$
- Exposure time can vary between 2ms - 30ms
- Clock frequency is 1 kHz
- The photodiode current is around 50pA in poor lighting, and around 750pA with good light conditions

2 Theoretical Background

2.1 Necessary background

To understand the further sections, the reader should be familiar with these concepts:

- MOS Transistors: Basic operation, body effect, subthreshold region, active region, leakage currents, active loads, NMOS/PMOS...
- Basic electronic components (from Figure 1.2)
- Schematic modeling tools: SPICE Models, Netlists...
- Basic circuit analysis, series Ohm law, charging/discharging of capacitors...
- Photography theory and ADC converters
- Hardware Description Languages: Digital system designing with Verilog
- Timing diagrams for digital systems simulation/validation

2.2 Acronyms

Hereafter the following acronyms are used so as to make the document more readable.

Acronym	**Description**
ADC	Analog Digital Converter
FSM	Finite State Machine
HDL	Hardware Description Language
I_D	Drain Current
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NMOS	Negative-channel Metal-Oxide Semiconductor
PMOS	Positive-channel Metal-Oxide Semiconductor
V_{DS}	Drain Source Voltage
V_{eff}	Effective voltage
V_{GS}/V_{SG}	Gate Source (Source Gate) Voltage
V_{OUT}	Output Voltage
V_{TN0}	Threshold voltage

Table 2.1: Acronyms used in this document

3 Design

This chapter shows how every designing choice is based on theory following a Top-Down design approach as for the description of the problem, for both the analog and digital circuits.

3.1 Analog system

Functional view of the readout of a pixel:

$$n \xrightarrow{conversion} i(n) \xrightarrow{charge} V_{CS}(n) \xrightarrow{amplify} V_{ADC}(n) \xrightarrow{quantize} b(n)$$

Where n is a scalar value representing the intensity of light arriving to the photodiode, i(n) is the current intensity produced by the same photodiode, $V_{CS}(n)$ is the voltage across the sampling capacitor, $V_{ADC}(n)$ is the input voltage of the ADC, and $b(n)$ is the binary number produced by the ADC.

3.1.1 Current conversion in photodiode

It is assumed that the photodiode exhibits an excellent linearity with respect to the incident light:

$$i(n) = conversion(n) = a_1 n + b_1 \tag{3.1}$$

Where b_1 is the so called dark current [1]

3.1.2 Capacitor charge

The voltage at the sampling capacitor can be formulated as follows [3]:

$$V_{CS}(n) = charge(i(n)) = \frac{Q(t)}{C} = \frac{1}{C}\int_{t_0}^{t} i(n)dt + V_{CS}(t_0) = \frac{t}{C}i(n) = \frac{a_1 t}{C}n + \frac{b_1 t}{C} \tag{3.2}$$

C_S is supposed to be fully discharged using the Erase signal before it is charged during an exposure time t. Thus $V_{CS}(t_0) = 0$ at $t_0 = 0$ and $t \in [2, 30]ms$

The formula shows that the charge is linear as long as the exposure time is constant. However, for different light conditions the user may change the exposure time to gain contrast in the image, as seen in Figure 3.1.

Figure 3.1: Use of exposure to get higher contrast in dark conditions [4]

Thus one could rewrite the previous expression as:

$$V_{CS}(n) = t(\sigma(N))(\frac{a_1}{C}n + \frac{b_1}{C}) = g(n)f(n) \tag{3.3}$$

Where $\sigma(N)$ represents the standard deviation derived from all n at every pixel.

So far $V_{CS}(n)$ holds a true proportional representation of the analog world because 3.2 is linear, and the non-linearity of 3.3 is between different images, maintaining the linearity between different pixels for every image. However, the approach wants to achieve linearity between b(n) and n because a change in the brightness should result in a proportional change in its binary representation, in other words, an unaltered representation of the analog world.

3.1.3 Signal amplification

Assuming that $V_{CS}(n)$ is linear from the previous section, this stage should achieve the following behaviour ($V_{OUT} = V_{ADC}$):

$$\frac{\partial V_{OUT}}{\partial V_{CS}} = constant \tag{3.4}$$

It is possible to simplify the circuit and approximate the calculus by treating the transistor M4 as an ideal switch and ignoring the transient nonlinear effect of the parasitic capacitor CC. Thereby one only have to consider the amplifying transistor M3 and the active load MC in series (cable N3 becomes merged with OUT). In order to satisfy condition 3.4 it is necessary that the circuit does not behave non-linearly, in other words, both transistors should work in the active region. This requirement implies the following equation system:

$$Safety\ Voltage \leq\ V_{eff}^{M3} \leq \frac{1}{2\theta} - Safety\ Voltage \tag{3.5}$$

$$Safety\ Voltage \leq\ V_{eff}^{MC} \leq \frac{1}{2\theta} - Safety\ Voltage \tag{3.6}$$

$$V_{OUT} >= V_{eff}^{M3} \tag{3.7}$$

$$V_{eff}^{M3} = V_{SG} - (V_{tn0} - \gamma(\sqrt{|2\phi_F| + V_{BS}} - \sqrt{|2\phi_F|})) \tag{3.8}$$

$$V_{eff}^{MC} = V_{SG} - V_{tn0} \tag{3.9}$$

Substituting constants with typical 180nm MOSFET process values (Table 1.3, Table 1.5 in the book CJM [2]) and 100mV for safety voltage to stay in the active region:

$$0.1 \leq\ V_{eff}^{M3} \leq 0.4 \tag{3.10}$$

$$0.1 \leq\ V_{eff}^{MC} \leq 0.4 \tag{3.11}$$

$$V_{OUT} >= V_{eff}^{M3} \tag{3.12}$$

$$V_{eff}^{M3} = V_{OUT} - V_{CS} - (0.45 + 0.5(\sqrt{0.7 + (1.8 - V_{OUT})} - \sqrt{0.7})) \tag{3.13}$$

$$V_{eff}^{MC} = (1.8 - V_{OUT}) - 0.45 \tag{3.14}$$

3.1.4 Determining M1 and M2 transistor sizes

From the solution of the equations system 3.10 - 3.14, the maximum value for V_{CS} of 0.559V is obtained and the minimum can be 0V if one wants to hold the linearity in both transistors. It is important, due to the low intensity produced by the photodiode and the amplification effect, to take into account small currents such as leakage and sub-threshold operation currents.

> "Three leakage currents are important in MOS transistors: sub-threshold leakage, gate leakage, and junction leakage. Of these, sub-threshold leakage is often the largest. It results in a finite drain current even when the transistor is off. Gate leakage results from quantum-mechanical tunneling of electrons through very thin gate oxides, and can be significant in digital circuits and when large gate areas are used." p.49 [2]

Therefore the gate and junction leakage shall be neglected because they may provoke only a very small non-linear change in C_S. The gate area in the M3 transistor will not be the maximum possible anyways as one shall see afterwards.

In the worst case scenario if sub-threshold currents happen during the readout, the capacitor could be empty before the readout at the ADC happens, producing always a 0 pixel value, or C_S could become overcharged, producing always a MAX_BINARY pixel intensity value.

Secondly, it could be possible only to allow a small linear change during the readout, producing equally linear readouts from the ADC. However, the intensity at sub-threshold operation is exponential with respect to V_{CS}. So the intensity through the M1 transistor should be minimized when the sampling capacitor has been charged and ready to read.

$$\begin{aligned} I_D^{M1} &= I_{D0}\left(\frac{W_{M1}}{L_{M1}}\right) e^{\left(\frac{qV_{eff}^{M1}}{nkT}\right)} \\ &= I_{D0}\left(\frac{W_{M1}}{L_{M1}}\right) e^{\left(\frac{q(V_{GS}-V_{tn})}{nkT}\right)} \\ &= I_{D0}\left(\frac{W_{M1}}{L_{M1}}\right) e^{-\left(\frac{q(V_{CS}+V_{tn0}+\gamma(\sqrt{2\phi_F+V_{CS}}-\sqrt{2\phi_F}))}{nkT}\right)} \end{aligned} \tag{3.15}$$

$$I_D^{M2} = I_{OFF}^{M2} = I_{D0}\left(\frac{W_{M2}}{L_{M2}}\right) e^{\left(\frac{qV_{eff}^{M2}}{nkT}\right)} = I_{D0}\left(\frac{W_{M2}}{L_{M2}}\right) e^{-\left(\frac{qV_{tn}}{nkT}\right)} \tag{3.16}$$

The numerator in the exponent of I_D^{M1} increases approximately linearly with V_{CS} (more than linear due to the body-effect), so the intensity still grows exponentially decreasing

V_{CS} (NB: negative exponent, OFF NMOS $V_G = 0$). So the maximum value of V_{CS} should be the maximum allowed in order to minimize this non-linear charging effect.

However, it is important to notice while I_D^{M1} depends on V_{CS}, I_D^{M2} remains constant. In order to minimize both currents, non-linear charging from M1 and linear (constant) discharging from M2, the maximum transistor length and the minimum width for both transistors shall be selected. It can be seen from equations 3.15 and 3.16 that I_D^{M2} is always bigger than I_D^{M1}, unless $V_{CS} = 0$, when they are equal. This means the discharging effect is stronger than the charging, resulting in an overall discharge effect.

$$W_{M1} = 2\mu m$$
$$L_{M1} = 2\mu m$$
$$W_{M2} = 2\mu m$$
$$L_{M2} = 2\mu m$$

3.1.5 Determining sampling capacitance

Since it was determined before that the maximum V_{CS} will be 0.559V, it implies that the sampling capacitance is:

$$C_{CS} = \frac{750pA}{0.559V}2ms = \frac{50pA}{0.559V}30ms = 2.683pF \tag{3.17}$$

3.1.6 Determining M3 and MC transistor sizes

Since an approximated circuit model is being used, where M3 and MC are in series:

$$I_D^{M3} = I_D^{MC} \tag{3.18}$$

$$\left(\frac{V_{eff}^{M3}}{V_{eff}^{MC}}\right)^2 = \frac{P_{MC}}{P_{M3}} = Ratio(V_{OUT}, V_{CS}) \tag{3.19}$$

Following the notation:

$$P_{MX} = \frac{W_{MX}}{L_{MX}} \tag{3.20}$$

Eqn. 3.18 forces the ratio of effective voltages to be constant because the ratio of transistor size ratio is constant since physical sizes cannot change.

$$Ratio(V_{OUT}, V_{CS}) = \left(\frac{V_{OUT} - V_{CS} - (V_{tn0} - \gamma(\sqrt{|2\phi_F| + V_{BS}} - \sqrt{|2\phi_F|}))}{V_{DD} - V_{OUT} - V_{tn}}\right)^2 \tag{3.21}$$

So the gradient of Ratio should be zero in order to keep it constant:

$$\nabla Ratio(V_{OUT}, V_{CS}) = (\frac{\partial Ratio(V_{OUT}, V_{CS})}{\partial V_{OUT}}, \frac{\partial Ratio(V_{OUT}, V_{CS})}{\partial V_{CS}}) = (0, 0) \tag{3.22}$$

$$\begin{aligned} &\frac{\partial}{\partial V_{OUT}} \left[\left(\frac{V_{OUT} - V_{CS} - (0.45 + 0.5(\sqrt{(1.8 - V_{OUT}) + 0.7} - \sqrt{0.7}))}{1.8 - V_{OUT} - 0.45} \right)^2 \right] \\ &= \frac{0.5(-0.5\sqrt{2.5 - V_{OUT}} + V_{OUT} - V_{CS} - 0.03167)}{(1.35 - V_{OUT})^3 \sqrt{2.5 - V_{OUT}}} (-4V_{CS}\sqrt{2.5 - V_{OUT}} \\ &\qquad + 5.27332\sqrt{2.5 - V_{OUT}} + V_{OUT} - 3.65) \\ &= 0 \end{aligned} \tag{3.23}$$

$$\begin{aligned} &\frac{\partial}{\partial V_{CS}} \left[\left(\frac{V_{OUT} - V_{CS} - (0.45 + 0.5(\sqrt{(1.8 - V_{OUT}) + 0.7} - \sqrt{0.7}))}{1.8 - V_{OUT} - 0.45} \right)^2 \right] \\ &= -\frac{2(-0.5\sqrt{2.5 - V_{OUT}} + V_{OUT} - V_{CS} - 0.03167)}{(1.35 - V_{OUT})^2} \\ &= 0 \end{aligned} \tag{3.24}$$

The partial derivatives become zero when the numerators are zero. Solving this equation system and finding the intersection of Figure 3.2, the point in which the gradient is zero is $V_{CS} = 0.794V$ and $V_{OUT} = 1.36V$. From these values one obtains:

$$Ratio(1.36, 0.794) = 1.013 \times 10^{-10} \tag{3.25}$$

Which results in:

$$P_{MC} = 1.013 \times 10^{-10} P_{M3} = \frac{W_{MC}}{L_{MC}} = 1.013 \times 10^{-10} \frac{W_{M3}}{L_{M3}} \tag{3.26}$$

This result can be interpreted as the optimal ratio to achieve a linear performance. Basically it needs to fulfill $P_{MC} << P_{M3}$. The only way to approach this rule is to make the width of M3 maximum, the length of M3 minimum, and the width of MC minimum, the length of MC maximum.

The interpretation of the optimal ratio is the following. The result of the equation system tell us which Ratio satisfies an ideal linear circuit because in equation 3.19 channel-length modulation second order effects are ignored while the transconductance is linear in the active region. However, since V_{CS} is going vary across the valid active region

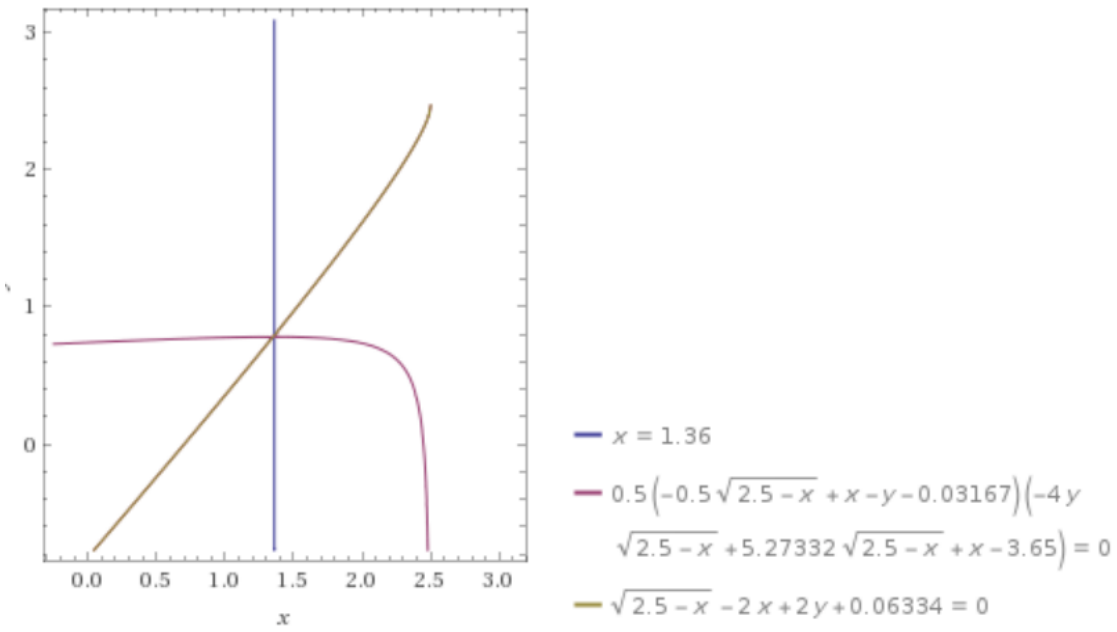


Figure 3.2: Graphic intersection when the derivatives equal zero

range, one sees that the ratio of effective voltage is not going to be constant because the gradient point is not in the range, and anyways, the range is wider than a single point. Then a nonlinear factor is introduced to keep the intensities at M3 and MC the same level.

$$\left(\frac{V_{eff}^{M3}}{V_{eff}^{MC}}\right)^2 \frac{(1+\lambda_{M3}(V_{DS}^{M3}-V_{eff}^{M3}))}{(1+\lambda_{MC}(V_{DS}^{MC}-V_{eff}^{MC}))} = \frac{P_{MC}}{P_{M3}} \tag{3.27}$$

So the final transistor sizes are:

$$W_{M3} = 10\mu m$$

$$L_{M3} = 0.7\mu m$$

$$W_{MC} = 2\mu m$$

$$L_{MC} = 2\mu m$$

So far M4 has been considered as an ideal switch, but its transistor size must also be set . M4 should have a minimum resistance when it's ON, in order to behave as closely as to a closed switch. Thus it should have the highest width/length ratio possible:

$$W_{M4} = 10\mu m$$

$$L_{M4} = 0.7\mu m$$

3.1.7 Voltage quantifying

Now that theoretically ADC has an approximate linear voltage input with respect to the brightness at the pixel photodiode, a linear ADC circuit can be used in order to translate linearly the light intensity to a binary number. Which is a solution to the main analog problem that it was defined previously in the introduction.

3.1.8 Analog subproblem solutions

A brief description of the subproblem solutions:

1. a) Every pixel circuit is the same.
   b) Eqn. 3.12 ensures that.

2. The sampling capacitance has been chosen so that a pixel of the image can have the maximum value when the light conditions are good and the exposure time is minimum. Eqn. 3.17

3. If one thinks of good light conditions as having a dynamic range of brightness across all the pixels, while the average is still high, then there is less probability to have overexposure because there is no need to increase the exposure time. To test the underexposure the voltage in poor light conditions and short exposure time can be checked:

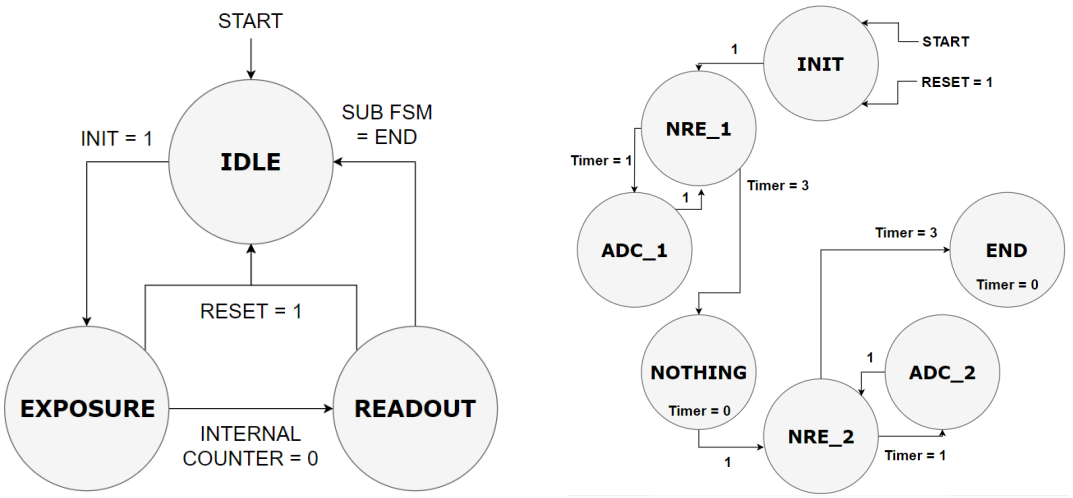
$$V_{CS} = \frac{50pA}{2.683pF} 2ms = 37.272mV \geq 0V \tag{3.28}$$

   If the ADC is able to translate any voltage from the active region range, then there is no possibility of underexposure.

3.2 Digital system

3.2.1 State diagrams

Changing to the digital system, once understood what the inputs and outputs do through all the states, two Finite State Machines (FSMs) are defined to meet the requirements:



(a) Main-FSM (b) Sub-FSM

Figure 3.3: Finite State Machines of the digital system

The main FSM (Figure 3.3a) perfectly matches the states that are requested for this project: IDLE, EXPOSURE and READOUT. The input "Init" makes the main FSM switch from IDLE to EXPOSURE state, and it stays there until "Internal counter" (light exposure time) becomes 0, going then to READOUT state. This last state requires a second FSM inside (sub-FSM), and the main FSM remains there until this sub-FSM finishes completely.

The sub-FSM (Figure 3.3b) defines the outputs activation timing, offering a synchronized approach with the states: INIT, NRE_1, ADC_1, NOTHING, NRE_2, ADC_2 and END. The states INIT, NOTHING and END are just simple transitions (so no outputs changes), but the rest activates the different outputs according to Figure 1.3.

In the main FSM, the outputs "Erase" and "Expose" are directly controlled. That would be for the left part of the analog circuit (Capacitor Cs charge and discharge). On the other hand, even thought the outputs "NRE_R1", "NRE_R2" and "ADC" are also activated/deactivated in such a FSM, its control is determined by the sub-FSM. That means they can only activate or deactivate in the state READOUT.

As both FSMs will work with the same clock signal, the synchronization between them will be absolute. In the same way, both of them will automatically go to the initial state if the reset signal is triggered, leaving the digital camera in stand-by one more time.

3.2.2 Block diagram

After the defined FSMs, Figure 3.4 shows the block diagram of the digital system. The module is called RE_control and is composed by 3 internal modules, each one of them doing a particular task:

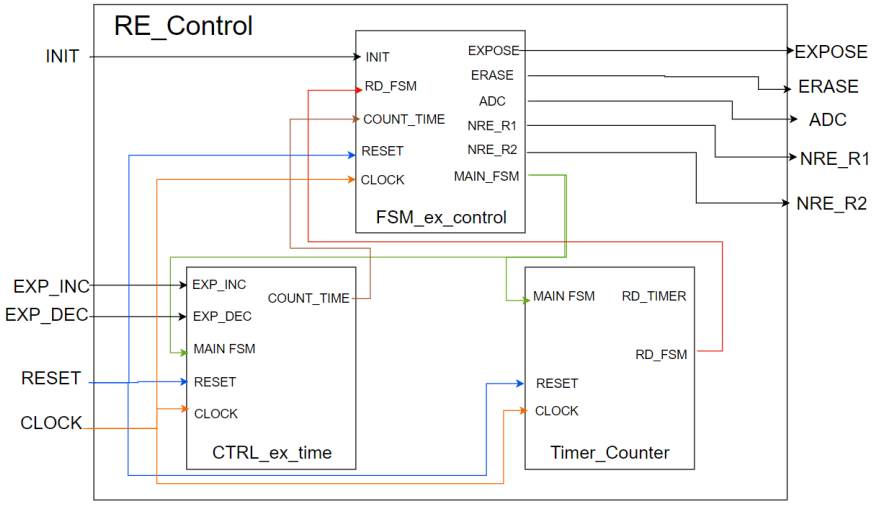


Figure 3.4: Block diagram of the digital system

- FSM_ex_control → The core of digital system. This module defines the main FSM [with the states IDLE, EXPOSURE and READOUT, explained before], contains the input "Init" to take a new picture as well as manages all the physical outputs.
- CTRL_ex_time → A support of the module FSM_ex_control. This module defines, through the inputs "Exp_increase" and "Exp_decrease" during the IDLE state, how long the exposure time will be later during the EXPOSURE state.
- Timer_Counter → A support of the module FSM_ex_control. Only working during the Readout state, this module defines the sub-FSM that the main module uses to determine the proper outputs activation. Only when the sub-FSM finishes, the main FSM will go to the IDLE state again.

Of course, all of them connect to the same clock and reset signals. The clock signal ensure synchronization among them, and the reset signal restarts the 3 sub-blocks at once. As an interesting point, the only element that is not synchronized is the input "reset", because it is considered an emergency signal (so it has priority the rest of the system), so it should not wait for the next rising edge of the "clock".

As mentioned before, the connection of these 3 modules leads to the complete digital system, called "RE_Control". The result is just one digital block that manages the whole analog design.

3.2.3 Verilog code

The Verilog code of the whole digital system is presented in the chapter 8 (Appendix). For the 4 modules (RE_control [Figure 7.4], FSM_ex_control [Figure 7.2], CTRL_ex_time [Figure 7.1], Timer_Counter [Figure 7.3]), the implementation as well its own testbench is shown.

To understand better the functionality of these program lines, please refer to the next subsection.

3.2.4 Description of the solution

A deep description of each block from Figure 3.4 is following explained:

- FSM_ex_control → With the maximum priority, the state in the main FSM will be IDLE if the "reset" signal is triggered.
  The IDLE state has only the output "Erase" activated, and the main FSM goes to EXPOSURE only when the input "Init"=1. This state will be hold as long as the input "count_time" (coming from CTRL_EX_TIME) is 1. Once it toggles to 0, the state changes finally to READOUT.
  Here the sub-FSM starts working (it is checked inside a different process), and even though it is managed by the module TIMER_COUNTER, its state is also shared as an input to this module, to decide how to act in consequence. For the sub-FSM, the state NRE_1 activates the output with the same name (the first two group of pixels) and the state ADC_1 activates the ADC in conjunction with this previous output as well. Exactly the same logic applies for the state NRE_2, but this time working with the second group of pixels.
  When this sub-FSM arrives to the end (state END), the main FSM goes back from READOUT to IDLE states to wait for performing the new picture. In addition, any kind of corruption during the process will lead to go back to the state IDLE.

- CTRL_ex_time → This module receives the main FSM status and it only works when it is either IDLE or EXPOSURE. In the IDLE state, the inputs "Exp_increase" and "Exp_decrease" let the counter "count_time" increase and decrease (in one unit, only in the rising edge) respectively, with a minimal value of 2ms and maximal value of 30ms.
  Once the block FSM_ex_control changes its state into EXPOSURE, this module subtracts one unit per clock signal to this counter, and outputs a binary signal with the same name (equivalent to the exposure time) as long as this counter is not zero yet. Once the counter reaches 0, the main FSM in FSM_ex_control will change into the last state READOUT.
  With the maximum priority, the counter "count_time" will be set to the minimum value (2ms) if the "reset" signal is triggered.

- Timer_Counter → This module defines the second (sub) FSM, which only works with the state READOUT in the main FSM. The sub-FSM moves through all states (so the activation/deactivation of all outputs) are completely managed just by time through the timer "RD_timer".
  The states NRE_1 and NRE_2 last 3ms (3 clock cycles) and, with the states ADC_1 and ADC_2, the ADC converter 1ms (1 clock cycle) in the middle of each NRE_X activation. The rest of the states are just transitions to ensure that the time between states is also fulfilled.
  With the maximum priority, the second FSM will go to the initial state if the reset signal is activated.

- RE_Control → The major module is basically composed by the previous 3 modules, so it just have the proper connections between them in order to provide a unique block.

Note: All these modules work inside a process each, activated in every rising edge of the clock signal (so synchronized). The only exception is the main module FSM_ex_control, which (for readable purposes) additionally has a second process in order to check (not to modify!) the state of the sub-FSM.

4 Simulations

This chapter shows all simulations done, used to validate the previous design. The analog system has been simulated with AIM SPICE, and the digital system with the online portal `https://www.edaplayground.com/`.

4.1 Analog system

Before running the simulations, it is important to change the gate oxide and threshold voltage parameters inside the provided file p18_cmos_models_tt.inc in order to match our model 180nm constants:

.param proc_delta = 0.82
.param vt_shift = 0.06

4.1.1 Sampling capacitor voltage

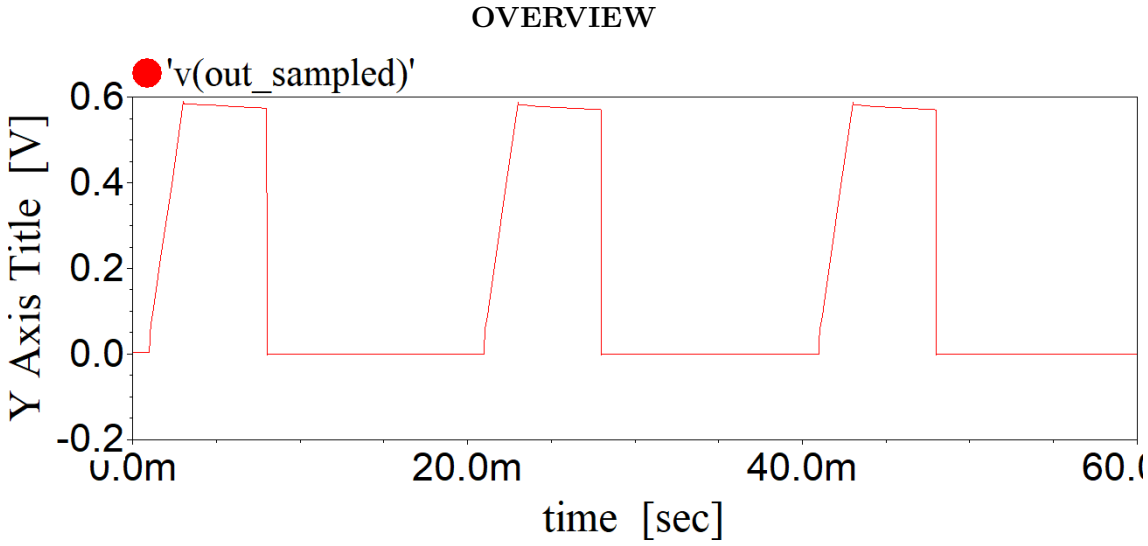


Figure 4.1: V_{CS} charge and discharge under 750pA and 2ms exposure time

The plots from this section belong to the testbench "Single.cir" which analyzes the behaviour of a single pixel. As Figure 4.1 shows, the readout process is repeated 3 times in a period of 60ms.

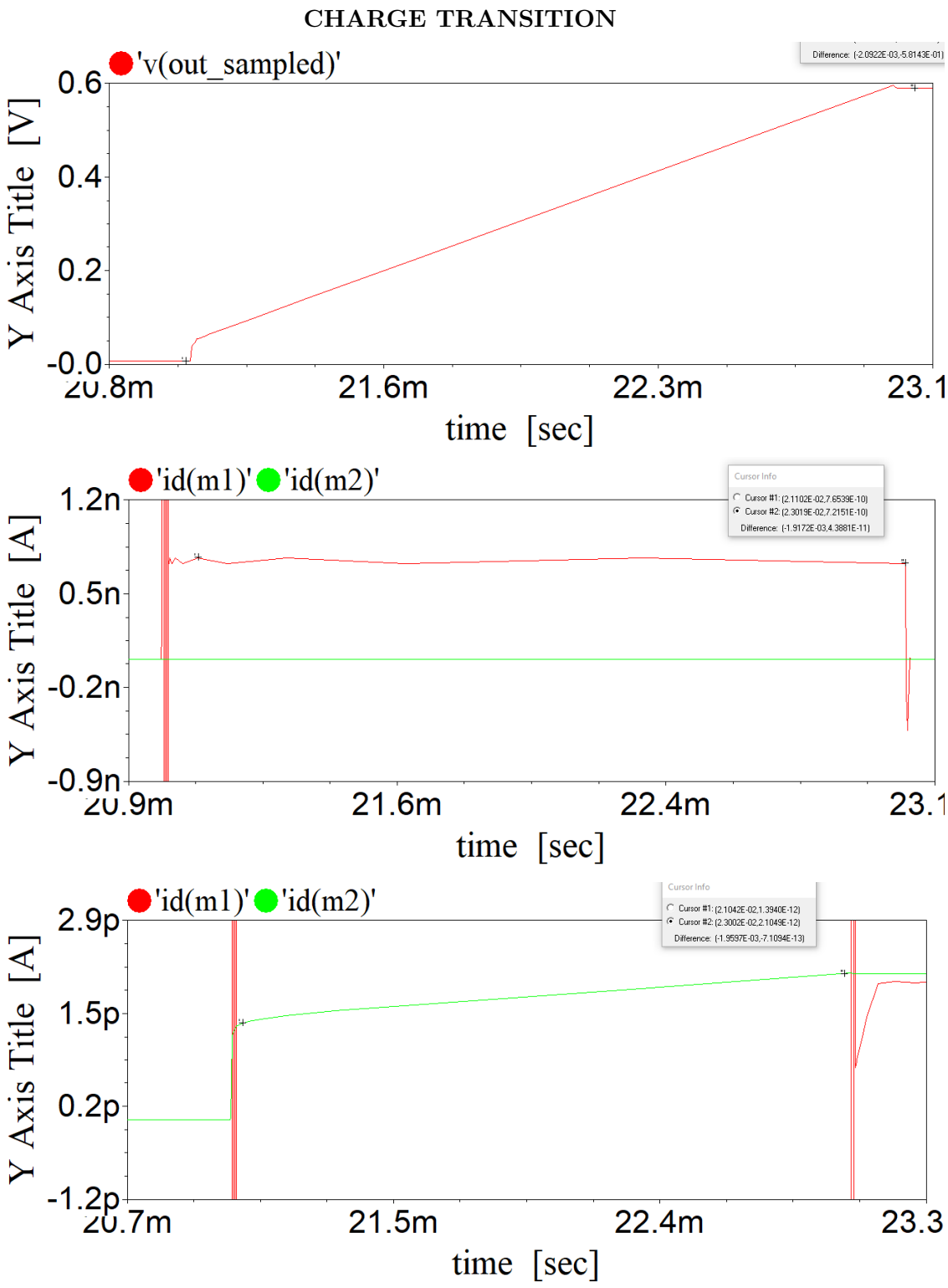


Figure 4.2: Transition when Expose signal is ON

From Figure 4.2 the linear prediction of the charge stage of V_{CS} made (by eqn. 3.2) can be verified. However, it can be seen at the beginning of the line that it is very steep before becoming linear. This effect is produced by the switch transistor M1 when the transition from OFF to ON causes a peak in the intensity when the transition is half-way. This effect was not taken into account in the design. Fortunately, since a safety voltage of 100mV for the active region has been chosen, the circuit will probably still work as modeled.

Another unexpected result is to see that I_D^{M2} is not constant with respect to V_{CS} as modeled in eqns. 3.16. This effect can be explained by the following book sentence:

> "Not captured here is drain-induced barrier lowering, a short-channel effect that causes subthreshold current to also depend on drain-source voltage." p.42 [2]

However, since this subthreshold current appears to be linear with a positive slope, it may not be such a drawback.

One can see from Figure 4.3 that the voltage drop is linear. This is in part because our the model approximately predicted that when V_{CS} is higher than 0, the linear I_D^{M2} would be higher than the non-linear I_D^{M1}, diminishing the non-linear component. A tiny linear slope can be seen in I_D^{M2} confirming the statement claimed by the book. Furthermore, one can observe a deep voltage drop right during the Exposure signal transition from ON to OFF. This effect is similar to the one seen in Fig 4.2 and may contribute to the circuit non-linearity since the voltage drop of in this figure is not inhibited by the voltage increase commented in Fig 4.2.

Figure 4.4 shows that it only takes $5.3\mu s$ to discharge the capacitor when it was charged to the maximum V_{CS} allowed. The intensity peaks show once again the effects of switching transistors from ON/OFF states.

Figure 4.5 shows a particularly important result. It shows that no matter which voltage has V_{CS} in the active region range, it will become nearly 0 after a certain period of time, keeping the system stable because the overall leakage intensities drag it down to nearly 0. In this case, one can notice the effect of the signal Erase from ON to OFF and the following transition to an stable "empty" V_{CS}. However, the second plot shows that the charging intensity I_D^{M1} is stronger than the discharging I_D^{M2} intensity. The only explanation to this fact may be heat dissipation, or other leakage factors not taken into account.

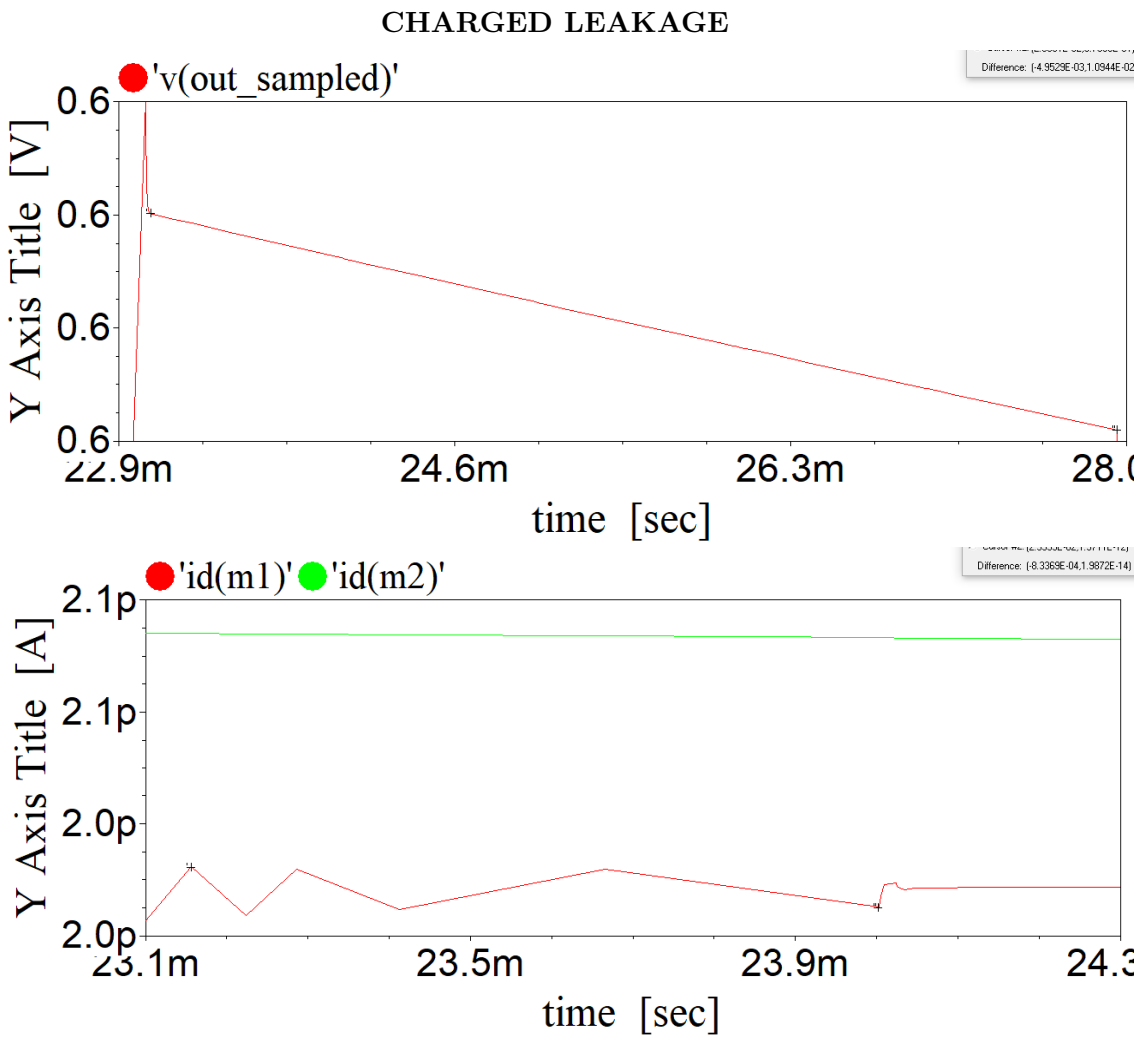


Figure 4.3: Leakage before readout

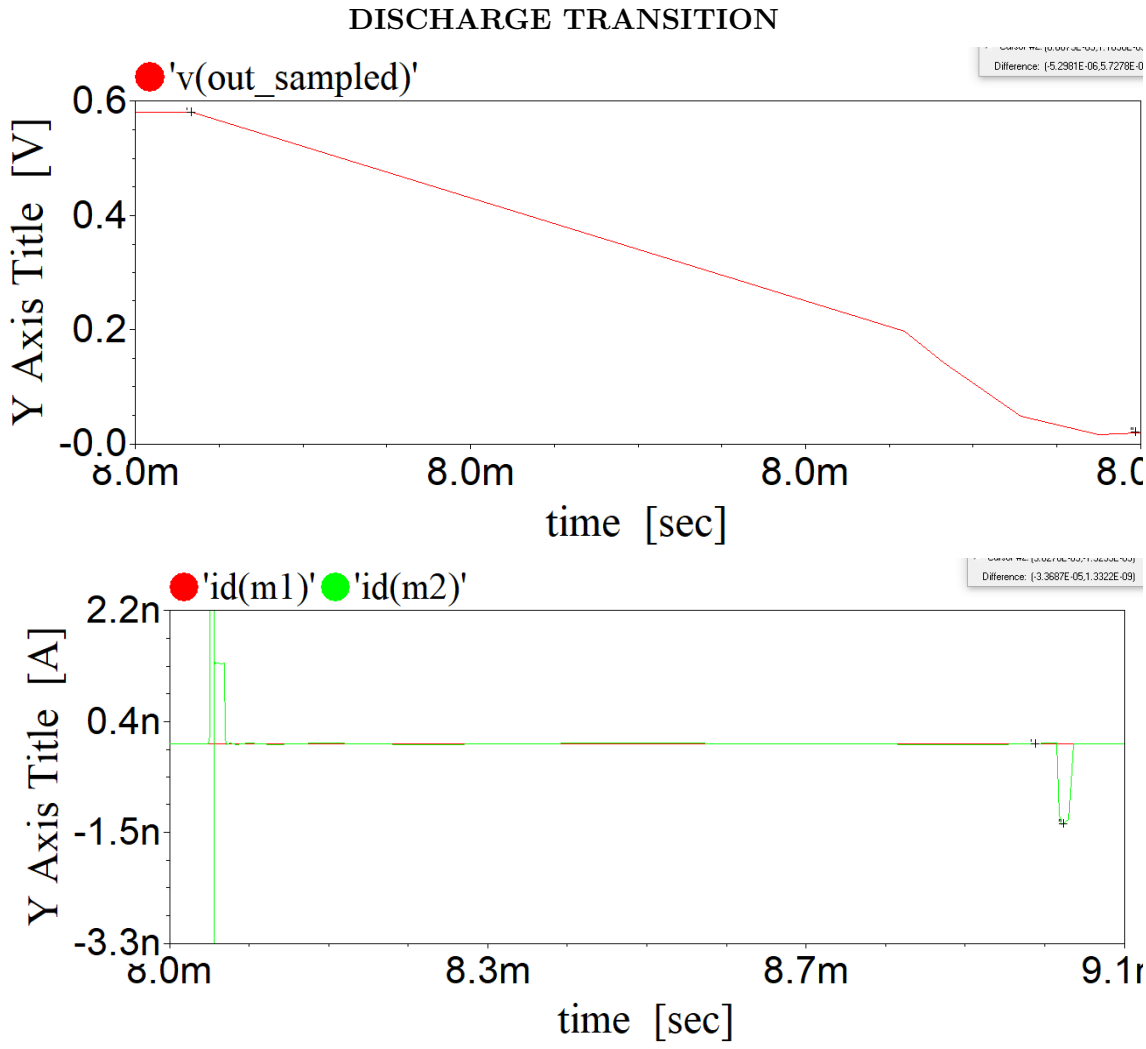


Figure 4.4: Transition when Erase signal is ON

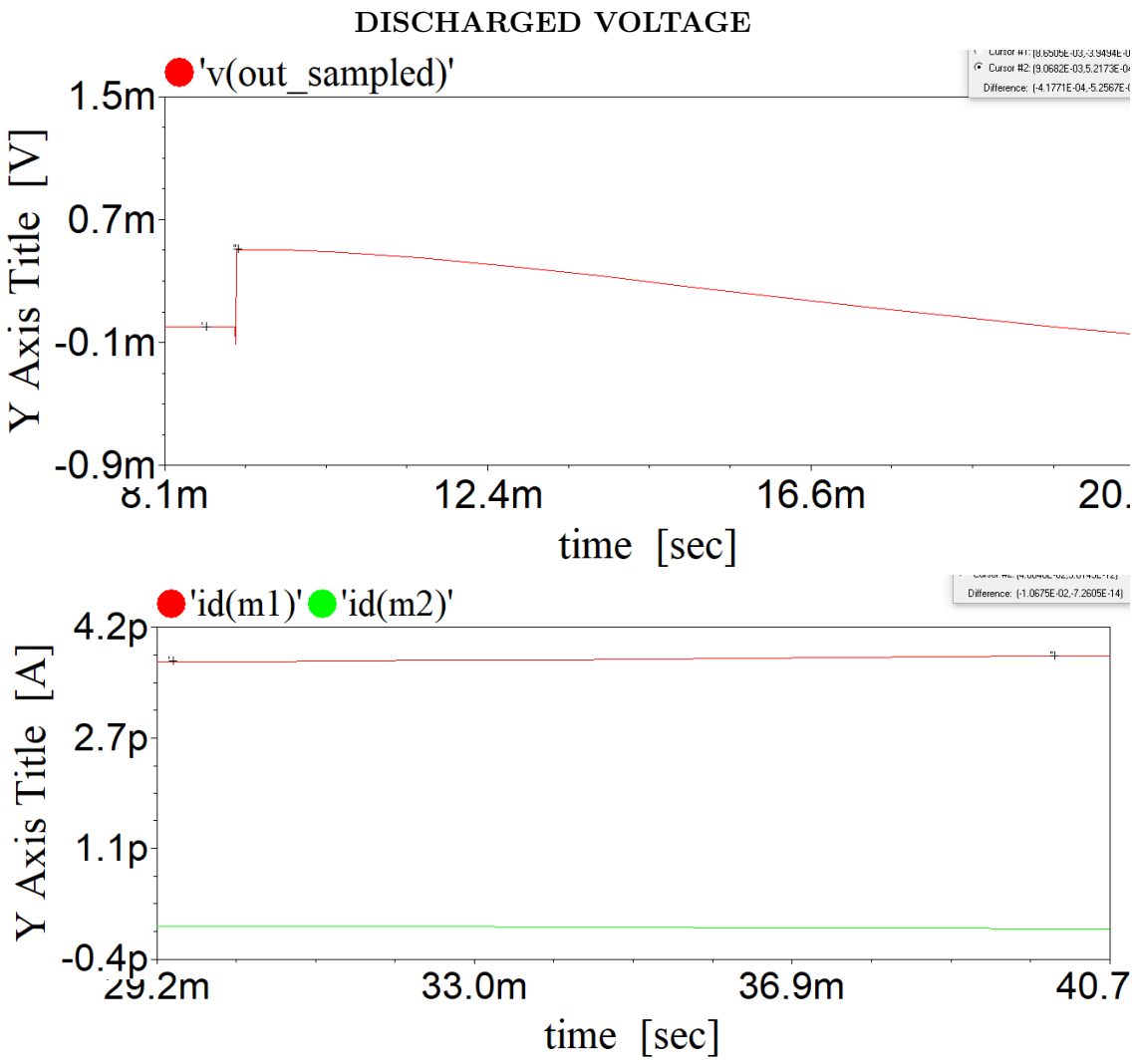


Figure 4.5: Low charge stability

4.1.2 ADC voltage

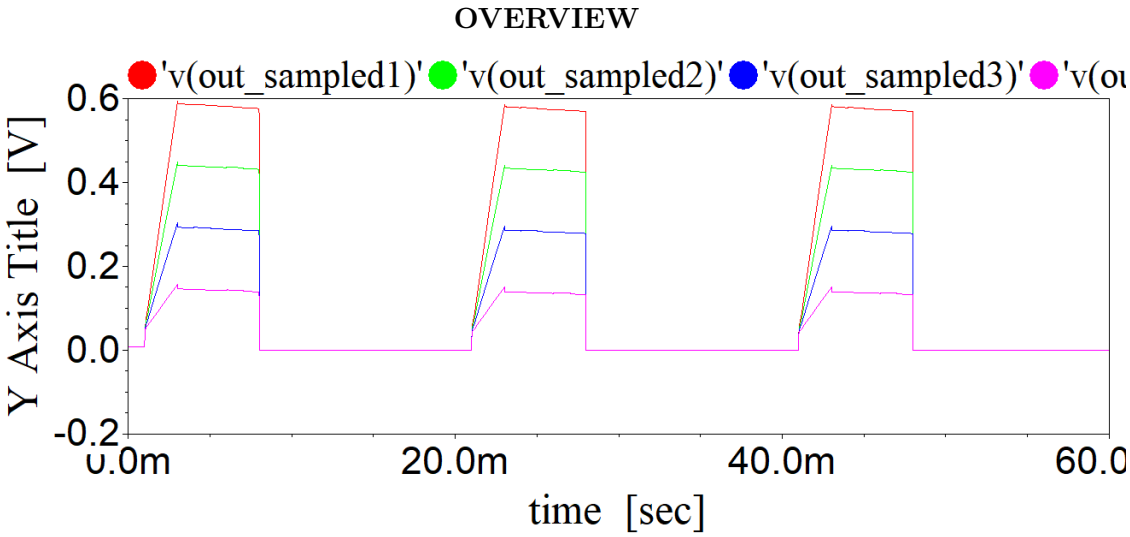


Figure 4.6: V_{CS} charge and discharge under 750pA, 550pA, 350pA, 150pA, and 2ms exposure time

The plots from this section belong to the testbench "Testbench.cir" which analyzes the behaviour of a 4 pixels (the final circuit). As Figure 4.1 shows, the readout process is repeated 3 times in a period of 60ms.

Figure 4.7 is probably the most decisive simulation of the analog part because it shows that the circuit satisfies eqn. 3.4 with a very high degree of accuracy, solving the main analog problem. Using the cursor tool from AIM Postprocessor, with 2ms of exposure the following light intensity is obtained -> VOUT map:

$$750pA- > 1.205V$$

$$550pA- > 1.122V$$

$$350pA- > 1.039V$$

$$150pA- > 0.954V$$

The reader can check that a difference in the input (200pA) causes a proportional difference in the output (0.084V) approximately. This is a great approximation after all the simplifications made in the model. The reader can also check that different exposure times hold quite well the linearity.

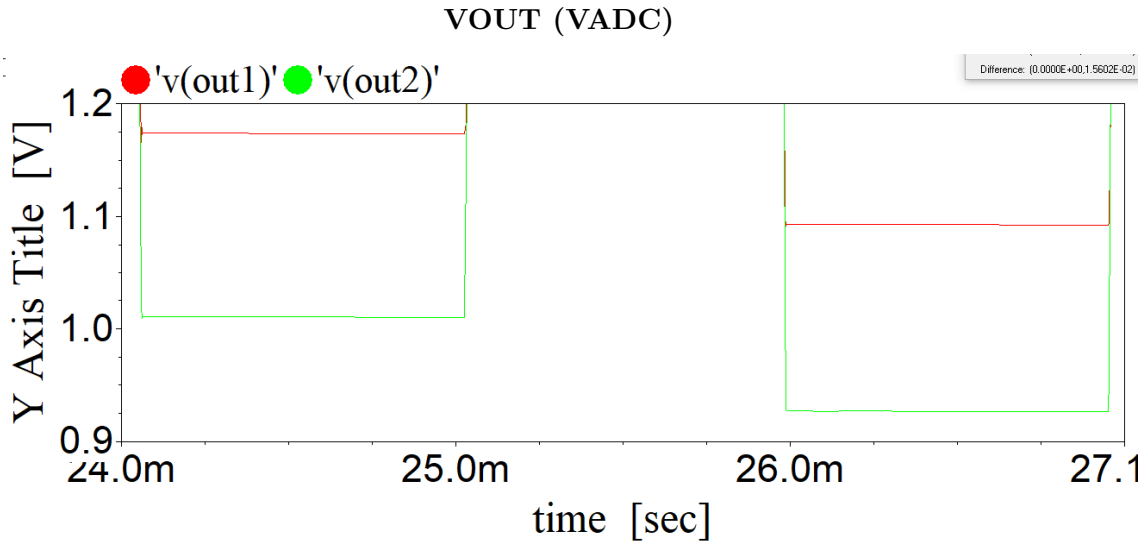


Figure 4.7: V_{CS} charge and discharge under 750pA, 550pA, 350pA, 150pA, and 2ms exposure time

4.2 Digital system

Using the Verilog code and testbenchs of all modules, the proper simulations are also done in order to validate the design according to the requirements.

Not only the inputs and outputs are shown here, but internal signals (as well as the parameters) are displayed too. In this way, verification can be done much easier.

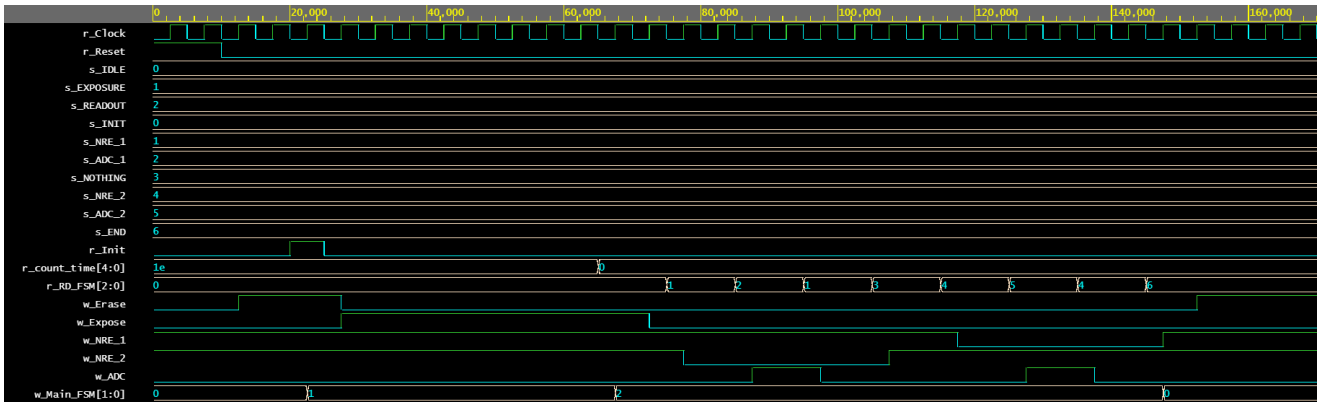


Figure 4.8: Simulation of FSM_ex_Control

Being "w_Main_FSM" the current state of the main FSM, and with all possible states the upper half numbered, the simulation in Figure 4.8 shows that the process starts when the input "r_Init" is pushed, inverting the logical states of "w_Erase" and "w_Expose" for a specific time and then the requested sequence for the pixels and ADC converter activation is properly executed.

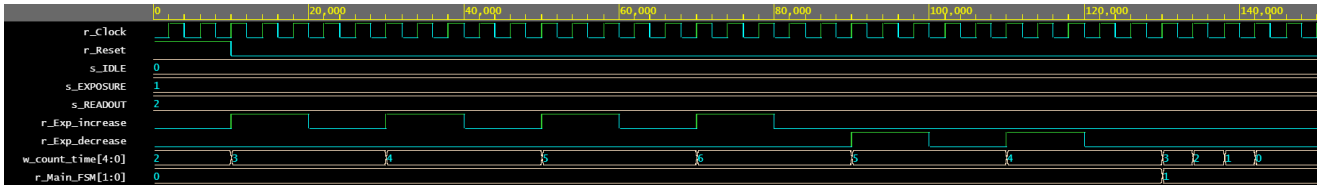


Figure 4.9: Simulation of CTRL_ex_time

The simulation of Figure 4.9 shows the counter for the exposure time "w_count_time" being increased and decreased according to inputs "r_Exp_increase" and "r_Exp_decrease" during the state IDLE, and once the state is EXPOSURE, the counter is reduced one unit per one clock cycle.

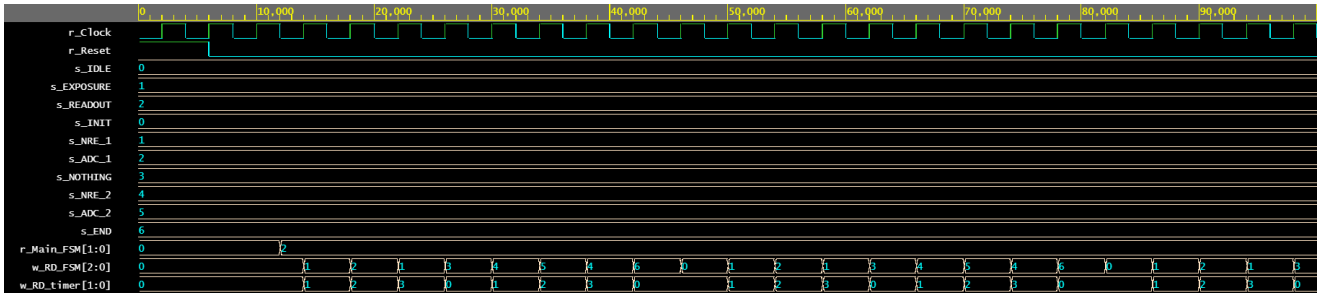


Figure 4.10: Simulation of Timer_Counter

Figure 4.10 simulation shows the correct sequence of the states in the sub-FSM, which is later used by FSM_ex_Control to update the right values of the outputs.

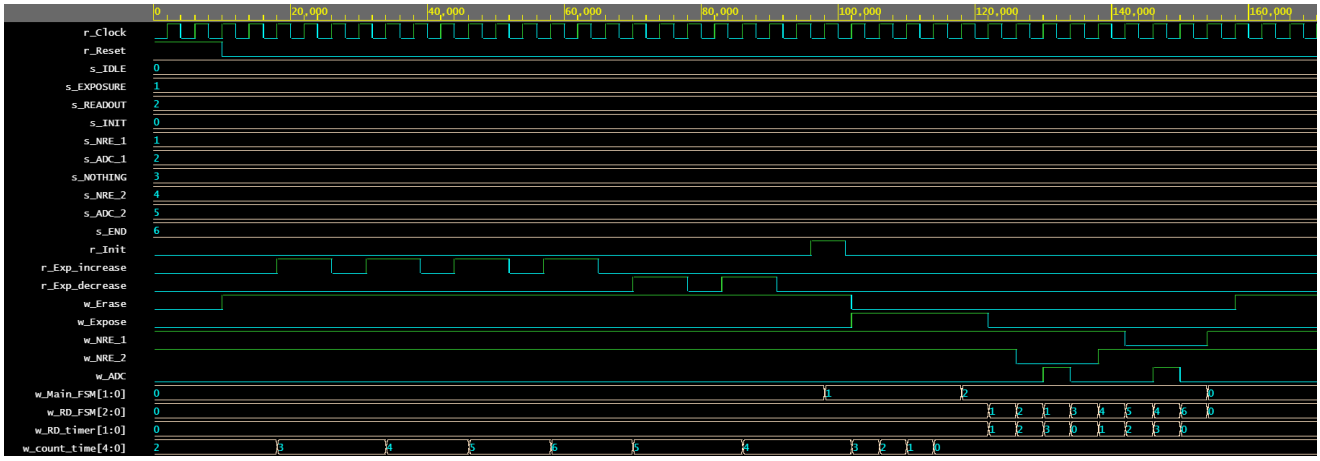


Figure 4.11: Simulation of RE_Control (complete digital system)

Figure 4.11 shows the final result, and one can say it is satisfactory because it matches with the requested one, shown in Figure 4.11. Figure 4.8, figure 4.9 and figure 4.10 show the individual testing of modules FSM_ex_control, CTRL_ex_time and Timer_Counter, respectively.

5 Results and discussions

The objectives of this project have been reached. The results in the chapter Simulations show us that the design behaves as expected, so that all requirements have been fulfilled.

Thanks to the easier and more efficient techniques to develop reliable digital systems, the response (timing diagram) of the digital part is almost perfect (as the main points in this kind of systems are the precision and robustness. Thus it is obvious to say that this part was the easiest.

The formulas and concepts seen during the semester have been useful to determine the parameters of the analog circuit. Such formulas are however an approximation of the real behaviour of the transistors, which results in some differences with respect to the theoretical model, as seen in the simulations.

6 Conclusions and future work

Analog and Digital electronics are quite different each other, but they work together since the moment that Digital (the later) was born, and we could see it within this project. Each part required a different knowledge and a different approach to be designed. And everything will be ok as long as the Digital part knows how and where to stimulate the Analog part. In addition, this project of the digital camera shows us that both kind of electronics will always exist together, as the digital one is much easier to work with, and the Analog one is mandatory to interact with our analog world.

We have learned how to do schematics from code (Netlists) as well as doing simulations and being able to understand what the displayed signals mean (the most important part from our point of view).

The Digital part was easier because there are many powerful tools to generate digital systems quite fast, easy simulations with just Boolean values and no physics are required to understand what is happening. Anyway, that does not mean it is useless at all. Hardware Description Languages (Verilog and VHDL) are the current and future trend of the industry for designing any kind of computer based electronics and it provides an endless number of possibilities to create and control technology.
We have learned the basics of a new HDL language (Verilog) and a very powerful approach to control/manage other devices: the Finite State Machine. We are pretty sure that these techniques will be very helpful as they are currently being used in the industry.

Overall, we enjoyed the project and it was really interesting for our background. All requirements were met and the results quite satisfactory.

Unfortunately, this project was just theoretical. We wanted to go a little bit further to integrate both systems into a physical device in order to validate our design with something else than just simulations. That would be an excellent additional point as a future work.
We recommend this project for future students, hoping that, as us, they will learn a lot about electronics integration as well.

Bibliography

[1] Hamamatsu. Photodiode technical information. http://users.monash.edu.au/~erict/Resources/sensors/photodiode_technical_information.pdf.

[2] Kenneth W. Martin Tony Chan Carusone, David A. Johns. *Analog Integrated Circuit Design*. Wiley, 2012.

[3] Wikipedia. Capacitor. https://en.wikipedia.org/wiki/Capacitor.

[4] Michael Zhang. What the naked eye sees in the night sky compared to what the camera can capture. https://petapixel.com/2015/04/04/what-the-naked-eye-sees-in-the-night-sky-compared-to-what-the-camera-can-capture/.

7 Appendix

7.1 SPICE netlists

7.1.1 "Testbench.cir"

```
* This is a parametrized testbench for your pixel circuit array

* You should at least test your circuit with:
* - current of 50 pA and exposure time 30 ms
* - current of 750 pA and exposure time 2 ms

* Instructions
* Connect EXPOSE, ERASE, NRE_R1 and NRE_R2 at the right place
* Run a transient simulation with length 60 ms
* Make sure outputs of pixel circuits to ADC are called OUT1 and OUT2
* Make plots of output voltages to ADC (here called OUT1 and OUT2)
* The voltage across internal capacitor (any pixel) is also of interest (here called OUT_SAMPLED1)
* You should also plot the control signals EXPOSE, NRE_R1, NRE_R2 and ERASE

.include p18_cmos_models_tt.inc

.subckt PhotoDiode VDD N1_R1C1
d1 N1_R1C1 vdd dwell 1
.model dwell d cj0=1e-14 is=1e-12 m=0.5 bv=40
Cd1 N1_R1C1 VDD 30f
.ends

.subckt Pixel VDD EXPOSE ERASE NRE OUT OUT_SAMPLED N1
M1 N1 EXPOSE OUT_SAMPLED 0 NMOS W=2u L=2u
M2 OUT_SAMPLED ERASE 0 0 NMOS W=2u L=2u
CS OUT_SAMPLED 0 2.683pf
M3 0 OUT_SAMPLED N3 VDD PMOS W=10u L=0.7u
M4 N3 NRE OUT VDD PMOS W=10u L=0.7u
.ends


.param Ipd_1 = 750p ! Photodiode current, range [50 pA, 750 pA]
.param Ipd_2 = 550p ! Photodiode current, range [50 pA, 750 pA]
```

```
.param Ipd_3 = 350p ! Photodiode current, range [50 pA, 750 pA]
.param Ipd_4 = 150p ! Photodiode current, range [50 pA, 750 pA]
.param VDD = 1.8 ! Supply voltage
.param EXPOSURETIME = 2m ! Exposure time, range [2 ms, 30 ms]

.param TRF = EXPOSURETIME/100 ! Risetime and falltime of EXPOSURE and ERASE signals
.param PW = EXPOSURETIME ! Pulsewidth of EXPOSURE and ERASE signals
.param PERIOD = EXPOSURETIME*10 ! Period for testbench sources
.param FS = 1k; ! Sampling clock frequency
.param CLK_PERIOD = 1/FS ! Sampling clock period
.param EXPOSE_DLY = CLK_PERIOD ! Delay for EXPOSE signal
.param NRE_R1_DLY = 2*CLK_PERIOD + EXPOSURETIME ! Delay for NRE_R1 signal
.param NRE_R2_DLY = 4*CLK_PERIOD + EXPOSURETIME ! Delay for NRE_R2 signal
.param ERASE_DLY = 6*CLK_PERIOD + EXPOSURETIME ! Delay for ERASE signal

VDD 1 0 dc VDD
VEXPOSE EXPOSE 0 dc 0 pulse(0 VDD EXPOSE_DLY TRF TRF EXPOSURETIME PERIOD)
VERASE ERASE 0 dc 0 pulse(0 VDD ERASE_DLY TRF TRF CLK_PERIOD PERIOD)
VNRE_R1 NRE_R1 0 dc 0 pulse(VDD 0 NRE_R1_DLY TRF TRF CLK_PERIOD PERIOD)
VNRE_R2 NRE_R2 0 dc 0 pulse(VDD 0 NRE_R2_DLY TRF TRF CLK_PERIOD PERIOD)

IA 1 NA DC Ipd_1
IB 1 NB DC Ipd_2
IC 1 NC DC Ipd_3
ID 1 ND DC Ipd_4

x1 1 NA PhotoDiode
x2 1 NB PhotoDiode
x3 1 NC PhotoDiode
x4 1 ND PhotoDiode

x5 1 EXPOSE ERASE NRE_R1 OUT1 OUT_SAMPLED1 NA Pixel
x6 1 EXPOSE ERASE NRE_R2 OUT1 OUT_SAMPLED2 NB Pixel
x7 1 EXPOSE ERASE NRE_R1 OUT2 OUT_SAMPLED3 NC Pixel
x8 1 EXPOSE ERASE NRE_R2 OUT2 OUT_SAMPLED4 ND Pixel
```

```
MC1 OUT1 OUT1 1 1 PMOS W=2u L=2u
CC1 OUT1 0 3pf

MC2 OUT2 OUT2 1 1 PMOS W=2u L=2u
CC2 OUT2 0 3pf

.plot V(OUT1) V(OUT2) ! signals going to ADC
.plot V(EXPOSE) V(NRE_R1) V(NRE_R2) V(ERASE)
.plot V(OUT_SAMPLED1) V(OUT_SAMPLED2) V(OUT_SAMPLED3) V(OUT_SAMPLED4)
```

7.1.2 "Single.cir"

```
* This is a parametrized testbench for your pixel circuit array

* You should at least test your circuit with:
* - current of 50 pA and exposure time 30 ms
* - current of 750 pA and exposure time 2 ms

* Instructions
* Connect EXPOSE, ERASE, NRE_R1 and NRE_R2 at the right place
* Run a transient simulation with length 60 ms
* Make sure outputs of pixel circuits to ADC are called OUT1 and OUT2
* Make plots of output voltages to ADC (here called OUT1 and OUT2)
* The voltage across internal capacitor (any pixel) is also of interest (here called OUT_SAMPLED1)
* You should also plot the control signals EXPOSE, NRE_R1, NRE_R2 and ERASE

.include p18_cmos_models_tt.inc

.subckt PhotoDiode VDD N1_R1C1
I1_R1C1 VDD N1_R1C1 DC Ipd_1
d1 N1_R1C1 vdd dwell 1
.model dwell d cj0=1e-14 is=1e-12 m=0.5 bv=40
Cd1 N1_R1C1 VDD 30f
.ends

.param Ipd_1 = 750p ! Photodiode current, range [50 pA, 750 pA]
.param VDD = 1.8 ! Supply voltage
.param EXPOSURETIME = 2m ! Exposure time, range [2 ms, 30 ms]

.param TRF = EXPOSURETIME/100 ! Risetime and falltime of EXPOSURE and
ERASE signals
.param PW = EXPOSURETIME ! Pulsewidth of EXPOSURE and ERASE signals
.param PERIOD = EXPOSURETIME*10 ! Period for testbench sources
```

```
.param FS = 1k; ! Sampling clock frequency
.param CLK_PERIOD = 1/FS ! Sampling clock period
.param EXPOSE_DLY = CLK_PERIOD ! Delay for EXPOSE signal
.param NRE_R1_DLY = 2*CLK_PERIOD + EXPOSURETIME ! Delay for NRE_R1
signal
.param NRE_R2_DLY = 4*CLK_PERIOD + EXPOSURETIME ! Delay for NRE_R2
signal
.param ERASE_DLY = 6*CLK_PERIOD + EXPOSURETIME ! Delay for ERASE
signal

VDD 1 0 dc VDD
VEXPOSE EXPOSE 0 dc 0 pulse(0 VDD EXPOSE_DLY TRF TRF EXPOSURETIME
PERIOD)
VERASE ERASE 0 dc 0 pulse(0 VDD ERASE_DLY TRF TRF CLK_PERIOD PE-
RIOD)
VNRE_R1 NRE_R1 0 dc 0 pulse(VDD 0 NRE_R1_DLY TRF TRF CLK_PERIOD
PERIOD)
VNRE_R2 NRE_R2 0 dc 0 pulse(VDD 0 NRE_R2_DLY TRF TRF CLK_PERIOD
PERIOD)

x1 1 N1 Photodiode
M1 N1 EXPOSE OUT_SAMPLED 0 NMOS W=2u L=2u
M2 OUT_SAMPLED ERASE 0 0 NMOS W=2u L=2u
CS OUT_SAMPLED 0 2.683pf
M3 0 OUT_SAMPLED N3 VDD PMOS W=10u L=0.7u
M4 N3 NRE OUT VDD PMOS W=10u L=0.7u

MC OUT OUT 1 1 PMOS W=2u L=2u
CC OUT 0 3pf

.plot V(OUT) !signals going to ADC
.plot V(EXPOSE) V(NRE_R1) V(NRE_R2) V(ERASE)
.plot V(OUT_SAMPLED)
.plot id(M1) id(M2)
```

7.2 Verilog code

Verilog code of every module defined, in conjunction with its own testbench:

7.2.1 CTRL_ex_time

```
module CTRL_ex_time (i_Exp_increase, i_Exp_decrease, i_Clock, i_Reset,
                     i_Main_FSM, o_count_time);
    input i_Exp_increase;
    input i_Exp_decrease;
    input i_Clock;
    input i_Reset;
    input [1:0] i_Main_FSM;
    output [4:0] o_count_time;

    parameter s_IDLE        = 2'b00;
    parameter s_EXPOSURE    = 2'b01;
    parameter s_READOUT     = 2'b10;

    reg pre_increase = 1'b0;
    reg pre_decrease = 1'b0;

    reg [4:0] r_count_time = 5'd2;

    always @(posedge i_Clock) begin
        if (i_Reset == 1'b1) begin
            pre_increase = 1'b0;
            pre_decrease = 1'b0;

            r_count_time = 5'd2;
        end
        else begin
            case (i_Main_FSM)
                s_IDLE : begin
                    if (i_Exp_increase==1'b1 && pre_increase==1'b0) begin
                        if (r_count_time < 5'd30) begin
                            r_count_time <= r_count_time + 1'd1;
                        end
                        else begin
                            r_count_time <= 5'd30;
                        end
                    end
                    else if (i_Exp_decrease==1'b1 && pre_decrease==1'b0) begin
                        if (r_count_time > 5'd2) begin
                            r_count_time <= r_count_time - 1'd1;
                        end
                        else begin
                            r_count_time <= 5'd2;
                        end
                    end
                    pre_increase <= i_Exp_increase;
                    pre_decrease <= i_Exp_decrease;
                end
                s_EXPOSURE : begin
                    if (r_count_time > 1'b0) begin
                        r_count_time <= r_count_time - 1'd1;
                    end
                    else begin
                        r_count_time <= 5'b0;
                    end
                end
                default : begin
                    r_count_time <= 5'b0;
                end
            endcase
        end
    end //always@ (posedge i_Clock)

    assign o_count_time = r_count_time;

endmodule //CTRL ex time
```

(a) Verilog code

```
`timescale 1us/1ns

module tb_CTRL_ex_time;
    parameter s_IDLE        = 2'b00;
    parameter s_EXPOSURE    = 2'b01;
    parameter s_READOUT     = 2'b10;

    reg r_Exp_increase = 1'b0;
    reg r_Exp_decrease = 1'b0;
    reg r_Clock = 1'b0;
    reg r_Reset = 1'b0;
    reg [1:0] r_State = s_IDLE;
    wire [4:0] w_count_time;

    CTRL_ex_time UUT (
        .i_Exp_increase(r_Exp_increase),
        .i_Exp_decrease(r_Exp_decrease),
        .i_Clock(r_Clock),
        .i_Reset(r_Reset),
        .i_State(r_State),
        .o_count_time(w_count_time)
    );
    always #2 r_Clock <= !r_Clock;

    initial begin   //Initial block
        r_Reset <= 1'b1; #10
        r_Reset <= 1'b0;
        r_State <= s_IDLE;

        r_Exp_increase <= 1'b1; #10
        r_Exp_increase <= 1'b0; #10
        r_Exp_increase <= 1'b1; #10
        r_Exp_increase <= 1'b0; #10
        r_Exp_increase <= 1'b1; #10
        r_Exp_increase <= 1'b0; #10
        r_Exp_increase <= 1'b1; #10
        r_Exp_increase <= 1'b0; #10

        r_Exp_decrease <= 1'b1; #10
        r_Exp_decrease <= 1'b0; #10
        r_Exp_decrease <= 1'b1; #10
        r_Exp_decrease <= 1'b0; #10

        r_State <= s_EXPOSURE;
    end
endmodule   //tb_CTRL_ex_time
```

(b) Testbench

Figure 7.1: Verilog code of CTRL_ex_time

7.2.2 FSM_ex_control

```verilog
module FSM_ex_control (i_Init, i_Clock, i_Reset, i_count_time,
                       i_RD_FSM, o_NRE_1, o_NRE_2, o_ADC,
                       o_Expose, o_Erase, o_Main_FSM);
    input i_Init, i_Clock, i_Reset;
    input [4:0] i_count_time;
    input [2:0] i_RD_FSM;
    output o_NRE_1, o_NRE_2, o_ADC, o_Expose, o_Erase;
    output [1:0] o_Main_FSM;

    parameter s_IDLE     = 2'b00;
    parameter s_EXPOSURE = 2'b01;
    parameter s_READOUT  = 2'b10;

    parameter s_INIT    = 3'b000;
    parameter s_NRE_1   = 3'b001;
    parameter s_ADC_1   = 3'b010;
    parameter s_NOTHING = 3'b011;
    parameter s_NRE_2   = 3'b100;
    parameter s_ADC_2   = 3'b101;
    parameter s_END     = 3'b110;

    reg r_NRE_1     = 1'b1;
    reg r_NRE_2     = 1'b1;
    reg r_ADC       = 1'b0;
    reg r_Expose    = 1'b0;
    reg r_Erase     = 1'b0;
    reg [1:0] r_Main_FSM = s_IDLE;

    always @(posedge i_Clock) begin
        case (i_RD_FSM)
            s_NRE_1 : begin
                r_NRE_1 <= 1'b0;
                r_ADC <= 1'b0;
            end
            s_ADC_1 : begin
                r_NRE_1 <= 1'b0;
                r_ADC <= 1'b1;
            end
            s_NRE_2 : begin
                r_NRE_2 <= 1'b0;
                r_ADC <= 1'b0;
            end
            s_ADC_2 : begin
                r_NRE_2 <= 1'b0;
                r_ADC <= 1'b1;
            end
            default : begin
                r_NRE_1 <= 1'b1;
                r_NRE_2 <= 1'b1;
                r_ADC <= 1'b0;
            end
        endcase
    end //always@ (posedge i_Clock)

    always @(posedge i_Clock) begin
        if (i_Reset == 1'b1) begin
            r_Main_FSM <= s_IDLE;
        end
        else begin
            case (r_Main_FSM)
                s_IDLE : begin
                    r_Expose <= 1'b0;
                    r_Erase <= 1'b1;
                    if (i_Init == 1'b1) begin
                        r_Main_FSM <= s_EXPOSURE;
                    end
                end
                s_EXPOSURE : begin
                    r_Expose <= 1'b1;
                    r_Erase <= 1'b0;
                    if (i_count_time == 1'b0) begin
                        r_Main_FSM <= s_READOUT;
                    end
                end
                s_READOUT : begin
                    r_Expose <= 1'b0;
                    r_Erase <= 1'b0;
                    if (i_RD_FSM == s_END) begin
                        r_Main_FSM <= s_IDLE;
                    end
                end
                default : begin
                    r_Main_FSM <= s_IDLE;
                end
            endcase
        end
    end //always@ (posedge i_Clock)

    assign o_NRE_1  = r_NRE_1;
    assign o_NRE_2  = r_NRE_2;
    assign o_ADC    = r_ADC;
    assign o_Expose = r_Expose;
    assign o_Erase  = r_Erase;
    assign o_Main_FSM   = r_Main_FSM;

endmodule //FSM_ex_control
```

(a) Verilog code

```verilog
`timescale 1us/1ns

module tb_FSM_ex_control;
    parameter s_INIT    = 3'b000;
    parameter s_NRE_1   = 3'b001;
    parameter s_ADC_1   = 3'b010;
    parameter s_NOTHING = 3'b011;
    parameter s_NRE_2   = 3'b100;
    parameter s_ADC_2   = 3'b101;
    parameter s_END     = 3'b110;

    reg r_Init      = 1'b0;
    reg r_Clock     = 1'b0;
    reg r_Reset     = 1'b0;
    reg [4:0] r_count_time = 5'd30;
    reg [2:0] r_RD_FSM = s_INIT;

    wire w_NRE_1;
    wire w_NRE_2;
    wire w_ADC;
    wire w_Expose;
    wire w_Erase;
    wire [1:0] w_State;

    FSM_ex_control UUT (
        .i_Init(r_Init),
        .i_Clock(r_Clock),
        .i_Reset(r_Reset),
        .i_count_time(r_count_time),
        .i_RD_FSM(r_RD_FSM),
        .o_NRE_1(w_NRE_1),
        .o_NRE_2(w_NRE_2),
        .o_ADC(w_ADC),
        .o_Expose(w_Expose),
        .o_Erase(w_Erase),
        .o_State(w_State)
    );
    always #2.5 r_Clock <= !r_Clock;

    initial begin
        r_Reset <= 1'b1; #20
        r_Reset <= 1'b0; #20

        r_Init <= 1'b1; #5
        r_Init <= 1'b0; #60
        r_count_time <= 5'd0;

        r_RD_FSM <= s_INIT;     #10
        r_RD_FSM <= s_NRE_1;    #10
        r_RD_FSM <= s_ADC_1;    #10
        r_RD_FSM <= s_NOTHING;  #10
        r_RD_FSM <= s_NRE_2;    #10
        r_RD_FSM <= s_ADC_2;    #10
        r_RD_FSM <= s_END;      #10
    end
endmodule   //tb_FSM_ex_control
```

(b) Testbench

Figure 7.2: Verilog code of FSM_ex_control

7.2.3 Timer_Counter

```
module Timer_Counter (i_Clock, i_Reset, i_Main_FSM,
                      o_RD_FSM, o_RD_timer);
    input i_Clock;
    input i_Reset;
    input [1:0] i_Main_FSM;
    output [2:0] o_RD_FSM;
    output [1:0] o_RD_timer;

    parameter s_READOUT = 2'b10;

    parameter s_INIT    = 3'b000;
    parameter s_NRE_1   = 3'b001;
    parameter s_ADC_1   = 3'b010;
    parameter s_NOTHING = 3'b011;
    parameter s_NRE_2   = 3'b100;
    parameter s_ADC_2   = 3'b101;
    parameter s_END     = 3'b110;

    reg [2:0] r_RD_FSM = s_INIT;
    reg [1:0] r_RD_timer = 2'b00;

    always @(posedge i_Clock) begin
        if (i_Reset == 1'b1) begin
            r_RD_timer <= 2'b00;
            r_RD_FSM <= s_INIT;
        end
        if (i_Main_FSM == s_READOUT) begin
            r_RD_timer <= r_RD_timer + 2'b01;
            case (r_RD_FSM)
                s_INIT : begin
                    r_RD_FSM <= s_NRE_1;
                end
                s_NRE_1 : begin
                    if (r_RD_timer == 2'b01) begin
                        r_RD_FSM <= s_ADC_1;
                    end
                    else if (r_RD_timer == 2'b11) begin
                        r_RD_timer <= 2'b00;
                        r_RD_FSM <= s_NOTHING;
                    end
                end
                s_ADC_1 : begin
                    r_RD_FSM <= s_NRE_1;
                end
                s_NOTHING : begin
                    r_RD_FSM <= s_NRE_2;
                end
                s_NRE_2 : begin
                    if (r_RD_timer == 2'b01) begin
                        r_RD_FSM <= s_ADC_2;
                    end if (r_RD_timer == 2'b11) begin
                        r_RD_timer <= 2'b00;
                        r_RD_FSM <= s_END;
                    end
                end
                s_ADC_2 : begin
                    r_RD_FSM <= s_NRE_2;
                end
                s_END : begin
                    r_RD_timer <= 2'b00;
                    r_RD_FSM <= s_INIT;
                end
                default : begin
                    r_RD_timer <= 2'b00;
                    r_RD_FSM <= s_INIT;
                end
            endcase
        end
        else begin
            r_RD_timer <= 2'b00;
        end
    end //always@ (posedge i_Clock)

    assign o_RD_FSM = r_RD_FSM;
    assign o_RD_timer = r_RD_timer;

endmodule //Timer_Counter
```

(a) Verilog code

```
`timescale 1us/1ns

module tb_Timer_Counter;
    parameter s_IDLE     = 2'b00;
    parameter s_EXPOSURE = 2'b01;
    parameter s_READOUT  = 2'b10;

    reg r_Clock = 1'b0;
    reg r_Reset = 1'b0;
    reg [1:0] r_State = s_IDLE;

    wire [2:0] w_RD_FSM;
    wire [1:0] w_RD_timer;

    Timer_Counter UUT (
        .i_Clock(r_Clock),
        .i_Reset(r_Reset),
        .i_State(r_State),
        .o_RD_FSM(w_RD_FSM),
        .o_RD_timer(w_RD_timer)
    );
    always #2 r_Clock <= !r_Clock;

    initial begin
        r_Reset <= 1'b1; #10
        r_Reset <= 1'b0; #8
        r_State <= s_READOUT;
    end
endmodule   //tb_Timer_Counter
```

(b) Testbench

Figure 7.3: Verilog code of Timer_Counter

7.2.4 RE_Control

```v
`include "FSM_ex_control.v"
`include "Timer_Counter.v"
`include "CTRL_ex_time.v"

module RE_Control (IN_Init, IN_Clock, IN_Reset,
        IN_Exp_increase, IN_Exp_decrease, OUT_NRE_1,
        OUT_NRE_2, OUT_ADC, OUT_Expose, OUT_Erase,
        OUT_count_time, OUT_Main_FSM, OUT_RD_FSM,
        OUT_RD_timer);

    input IN_Init, IN_Clock, IN_Reset,
            IN_Exp_increase, IN_Exp_decrease;
    output OUT_NRE_1, OUT_NRE_2, OUT_ADC,
            OUT_Expose, OUT_Erase;

    output [4:0] OUT_count_time;
    output [1:0] OUT_Main_FSM;
    output [2:0] OUT_RD_FSM;
    output [1:0] OUT_RD_timer;

    wire [1:0] w_Main_FSM;
    wire [2:0] w_RD_FSM;
    wire [4:0] w_count_time;
    wire [1:0] w_RD_timer;

    FSM_ex_control block1 (
        .i_Init(IN_Init),
        .i_Clock(IN_Clock),
        .i_Reset(IN_Reset),
        .i_count_time(w_count_time),
        .i_RD_FSM(w_RD_FSM),
        .o_NRE_1(OUT_NRE_1),
        .o_NRE_2(OUT_NRE_2),
        .o_ADC(OUT_ADC),
        .o_Expose(OUT_Expose),
        .o_Erase(OUT_Erase),
        .o_Main_FSM(w_Main_FSM)
    );
    CTRL_ex_time block2 (
        .i_Exp_increase(IN_Exp_increase),
        .i_Exp_decrease(IN_Exp_decrease),
        .i_Clock(IN_Clock),
        .i_Reset(IN_Reset),
        .i_Main_FSM(w_Main_FSM),
        .o_count_time(w_count_time)
    );
    Timer_Counter block3 (
        .i_Clock(IN_Clock),
        .i_Reset(IN_Reset),
        .i_Main_FSM(w_Main_FSM),
        .o_RD_FSM(w_RD_FSM),
        .o_RD_timer(w_RD_timer)
    );
    assign OUT_count_time = w_count_time;
    assign OUT_Main_FSM = w_Main_FSM;
    assign OUT_RD_FSM = w_RD_FSM;
    assign OUT_RD_timer = w_RD_timer;

endmodule //RE_Control
```

(a) Verilog code

```v
`timescale 1us/1ns

module tb_RE_Control;
    reg r_Init = 1'b0;
    reg r_Clock = 1'b0;
    reg r_Reset = 1'b0;
    reg r_Exp_increase = 1'b0;
    reg r_Exp_decrease = 1'b0;

    wire w_NRE_1;
    wire w_NRE_2;
    wire w_ADC;
    wire w_Expose;
    wire w_Erase;

    wire [4:0] w_count_time;
    wire [1:0] w_Main_FSM;
    wire [2:0] w_RD_FSM;
    wire [1:0] w_RD_timer;

    RE_Control UUT (
        .IN_Init(r_Init),
        .IN_Clock(r_Clock),
        .IN_Reset(r_Reset),
        .IN_Exp_increase(r_Exp_increase),
        .IN_Exp_decrease(r_Exp_decrease),
        .OUT_NRE_1(w_NRE_1),
        .OUT_NRE_2(w_NRE_2),
        .OUT_ADC(w_ADC),
        .OUT_Expose(w_Expose),
        .OUT_Erase(w_Erase),
        .OUT_count_time(w_count_time),
        .OUT_Main_FSM(w_Main_FSM),
        .OUT_RD_FSM(w_RD_FSM),
        .OUT_RD_timer(w_RD_timer)
    );
    always #2 r_Clock <= !r_Clock;

    initial begin
        r_Reset <= 1'b1; #10
        r_Reset <= 1'b0; #8

        r_Exp_increase <= 1'b1; #8
        r_Exp_increase <= 1'b0; #5
        r_Exp_increase <= 1'b1; #8
        r_Exp_increase <= 1'b0; #5
        r_Exp_increase <= 1'b1; #8
        r_Exp_increase <= 1'b0; #5
        r_Exp_increase <= 1'b1; #8
        r_Exp_increase <= 1'b0; #5

        r_Exp_decrease <= 1'b1; #8
        r_Exp_decrease <= 1'b0; #5
        r_Exp_decrease <= 1'b1; #8
        r_Exp_decrease <= 1'b0; #5

        r_Init <= 1'b1; #5
        r_Init <= 1'b0; #60
    end

endmodule   //tb_RE_Control
```

(b) Testbench

Figure 7.4: Verilog code of RE_Control (complete digital system)