Analog/Digital front-end Design for a 4-pixel Digital Camera

Ahmed El-Yamany, Anar Aliyev

Abstract—Digital cameras acts as a perfect example for mixed-signals devices. Both digital and analog parts are implemented in parallel and are tested in a stand-alone fashion. In the analog design, we try to identify the trade-offs for the different circuits used, and maintain the functionality in the top level. In the digital design, we try to cover all the design specification for controlling the analog part and the specified wave-forms. We use a test plan to make sure we covered all the specs using directed stimuli testing. We summarize all the simulations, results and wave-forms in the following report.

Index Terms—Digital Camera, 4-pixels, Front-end, Verilog-HDL, AIM-SPICE.

1 Introduction

THIS project is intended to implement digital camera using both analog and digital circuits. Analog part includes photo-diode, which is a sensor to absorb the light from the analog world and convert it to proportional current. The analog part also includes switching transistors; ″expose″ switch for charging the sampling capacitor with current from the photo-diode, ″erase″ switch for releasing the charge from capacitor after every exposure cycle and ″nre″ switches for reading the voltage that we got on the capacitor. Furthermore, a buffer transistor, which will read voltage from capacitor without disturbing it. In series with an active load transistor connected in a diode-connection fashion.

The Digital part we will be discussing is the control logic for such aforementioned signals in the analog switches part. The digital block gets inputs to control its functionality as init, exp_dec, exp_inc, reset and clk input signals. Init signal is an input from camera trigger to activates the block as a new image is to be taken, exp_dec and exp_inc signals are used for decreasing and increasing exposure time respectively -user should use those with respect to the light conditions-, reset signals resets the whole system and clk for system clocking.

In our solution we have used AIM-SPICE for the analog part simulations and Active-HDL for designing the digital part. While designing analog circuitry, we tried to minimize the leakage current with respect to speed for the switching circuits. The parameters of source follower and load transistors were chosen to maintain their functionality along corner cases. On the digital side, We used a finite state machine (FSM) which includes 3 states : ″read-out″ for reading the output from analog circuit, ″exposure″ for taking the picture and an ″idle″ state.

This report is organized as follows. There are 2 major sections dividing our work into analog and digital parts. In the first major section, we discuss the analog part. We first start with each circuit analysis on its own as: switches, buffer and active load circuits. Finally we merge all of them to create the pixel circuit, we also introduce the calculations for the capacitor used in that part. The second major chapter discusses the digital part, we start with introducing the specification and requirement derived from the analog circuitry. In the next sub-chapter, we introduce our design, including the timing charts, FSM diagrams and block diagrams. Finally we conclude with the simulations,that ensured that the digital part meets all the specification and requirements.

2 Analog Circuit

The analog part acts as the main functionality circuit for the digital camera, the photo-diode gets the light and converts it to the current. The current flows to the analog circuitry through the different switches: expose switch, erase switch and NRE switch. To get the voltage from this current, we use a capacitor that is charged according to the expose time. Then using a source follower PMOS buffer, we get the voltage on the capacitor without discharging it. The output voltage is then transferred over the active load resistance to the next stage.

2.1 Switching Circuits

The switching circuits are the main part of the analog circuit. They are the connection between the digital input control signals and the analog circuit. The switches we have are the ″expose″, ″erase″ and ″nre″ switches. Each one controls a certain functionality in the analog circuit as described below.

Expose Switch The expose switch handles the input pulse signal from the digital part which actually mimics the push on the expose button by the camera user. The pulse has a restriction of maximum frequency of 0.5 kHz, as the shortest pulse is 2 ms long.

Erase Switch The capacitor that holds the charge of the last exposure needs to be refreshed after every exposure. The erase switch is used to enable a discharging outlet so that the capacitor can fully discharge. The erase switch has a constraint of 1 KHz frequency same as the whole digital circuitry working with 1 kHz clk.

NRE Switch Since we are designing the circuit for 4 pixels and we are only using 2 ADCs, we need to multiplex the input from each 2 pixel on the same row for each ADC input. We use the NRE switch also to do that, it has a constraint frequency of around 0.33 kHz, indicating a change within 3 clk cycles of 1 kHz clock.

As we notice from the analysis, that the speed constraint on the switching circuits is 1 kHz. Another constraint to take into consideration is the leakage current (also called sub-threshold current). The leakage current is the current that flows between source and drain of the mosfet, when the mosfet is off. The leakage current problem arises from the low output current expected from the photo-diode in dimmer situations. The current reaches a lower range of 50 pA, so regardless how small the leakage current is, it would still significantly affect the worst case current.

In equation 1, we show the relation between the mosfet dimensions and the leakage current, when the mosfet is used as a switch. eq. (1) shows the direct relation between the leakage current and the dimensions, if we considered a fixated V$_{eff}$ voltage, which is our case.

$$I_ds(nA) \propto \frac{W}{L}.e^{q(V_{gs}-V_t)/\eta KT} \tag{1}$$

In eq. (2), we can see how the speed also relates to the mosfet length and width, using the equation of delay. The delay is directly proportional to L/W, making speed directly proportional to W/L.

$$\tau = RC; \quad with\, R = \frac{L}{KWV_{eff}} \tag{2}$$

There is obviously a trade-off between speed and leakage current being the 2 constrains we are concerned about due to the specifications. The relationship between Speed and transistor dimensions is directly proportional, same with the leakage current. So the less leakage current, the less speed we can expect. Due to the lower speed limitation we have here, and due to the high limitation on the leakage current, we wanted to get the smallest leakage current possible, so we used the smallest dimension for W and highest for L with the specified limit to get the least $\frac{W}{L}$ possible, which is 1 at W = 2μ and L = 2μ.

In Table 1, we show the least leakage current values, we got when using W/L = 1. In Figure 1 we show the corresponding switching speed of the NMOS, by plotting the input and output at the max frequency with the worst input current of 50 pA.

If we doubled the $\frac{W}{L}$, we can see the leakage current behaves as we expected in Table 1, on the other hand in Figure 2 we can notice a slight increase in the switching speed. It is quite unremarkable because of the limited frequency of the input.

Table 1
Leakage Current with respect to the Mosfet Dimensions

$\frac{W}{L}$	Leakage Current (pA)
1	10
2	19

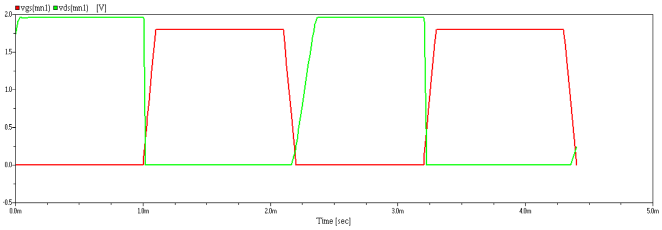


Figure 1. Switching Speed at W/L = 1 with 50 pA input current

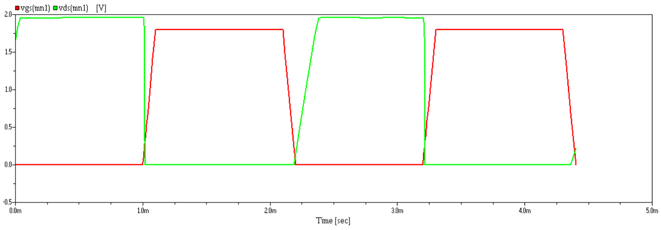


Figure 2. Switching Speed at W/L = 2 with 50 pA input current

2.2 Active Load Circuit

The output voltage needs an output resistance to be loaded, we use a PMOS in a diode connected configuration to work as a resistance. In fig. 3, we can see the schematic for this circuit. The DC analysis, as we see in fig. 3, so we will have the PMOS always working in the saturation region, also called perfect saturation.

$$V_g = V_d, \wedge V_{sg} = V_{dd}, V_{sd} > V_{eff} \tag{3}$$

In the small analysis model in fig. 3, we need to calculate the resistance that will be added to the buffer circuit connected in cascade.

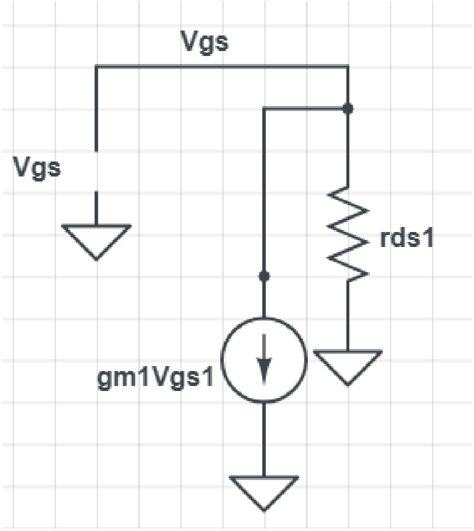


Figure 3. Small Signal Model for Active Load NMOS

$$R_{ds} = \frac{1}{g_{m1}}. \tag{4}$$

2.3 Buffer Circuit

The buffer circuit gets the voltage from "CS" capacitor and outputs it. Thus the parameters of the source follower should be calculated to get unity gain, additionally, the input impedance of the buffer should be really high and output impedance should be really low to avoid voltage distortions. To do this we had to analyze the small signal model for the buffer shown in fig. 4.

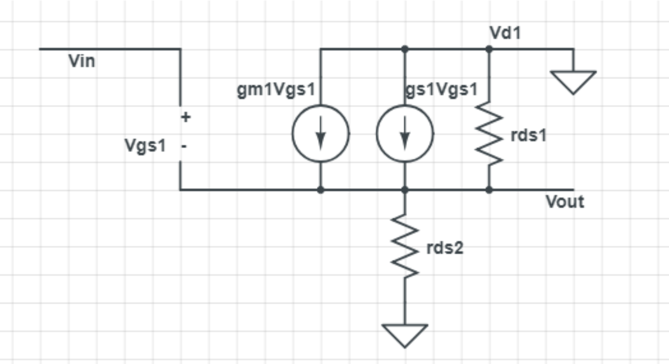


Figure 4. Small Signal Model for Buffer PMOS

Also the active load transistor should be taken into consideration. The resulting small signal model is shown in fig. 5.

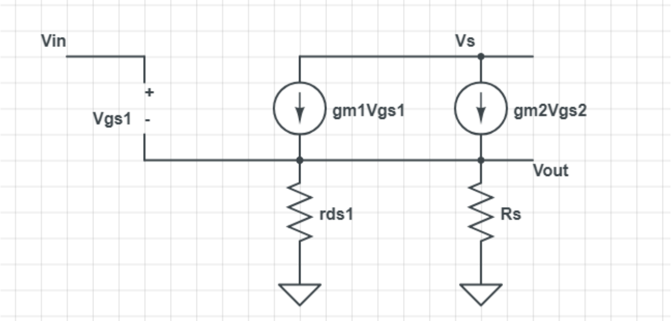


Figure 5. Small Signal Model for Buffer PMOS including Active Load

After simplifying the above small signal model, we get fig. 6.

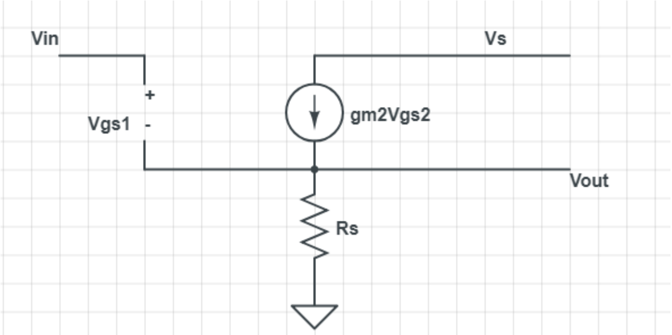


Figure 6. Simplified Small Signal Model for Buffer PMOS

$$A = G_m R_{out}; \tag{5}$$

eq. (5) tells us that we need to calculate the output resistance for the configuration. Consequently, just by sight, we can observe that the output impedance in the configuration in fig. 5 is just some resistances in parallel between the source node and the ground. When dealing with their trans-conductance, the parallel expression turns to a simple summation equation. In eq. (6), we show the expression for the total output resistance.

$$R_{out} = \frac{1}{g_{m1} + g_{s1} + g_{ds1} + g_{ds2}} \tag{6}$$

$$A = \frac{g_{m1}}{g_{m1} + g_{s1} + g_{ds1} + g_{ds2}} \tag{7}$$

To reach our target unity gain from the buffer, we need to equate the expression in eq. (7) with 1. Then get the optimum values for the different parameters g_{m1}, g_{s1}, g_{ds1}, g_{m2} that meet our criteria. To simplify this relation, we use 2 assumptions and we discuss them in the following points:

- g_{s1}: We assume here that the bulk effect for the buffer circuit would be low relatively to the g_{m2} and g_{m1}.
- g_{ds1}: Our experiments show that R$_{ds1}$ is in the range of 5 MΩ as shown in the figure, for different values of $\frac{W}{L}$. So, for the same reason as above, we approximately eliminate it from the expression to get an easier derivation.

We end up with the eq. (8) that would let us get the approximate dimensions for our buffer and load circuits to get unity gain.

$$A = \frac{g_{m1}}{g_{m1} + g_{ds2}}, \wedge\, g_{ds2} = g_{m2} \tag{8}$$

In eq. (8), we could reach unity gain, if g_{m2} = 0, or relatively very small compared to the g_{m1}. In the same sense, the smaller the $\frac{W}{L}$ for the load, and the higher for the buffer, the more the gain expression would approach unity. We illustrate this in eq. (9).

$$A = \frac{g_{m1}}{g_{m1}} = 1, \wedge\, g\_m1 >>> g\_m2 \tag{9}$$

2.4 Top Level Analog Circuit

In fig. 7, we take a look on the top level for the analog part in our digital camera. We have already gone through most of the circuit in the previous chapters. In the top level, we only introduce the photo-diode and the capacitor among the previously mentioned configurations as the 3 switches, the buffer and load circuit.

The photo-diode as discussed in the intro. gets the input light for this pixel and converts the light to the current within the specified range from 50 pA to 750 pA. The sampling capacitor on the other hand is used to convert the current to voltage. The expose signal is used to charge the capacitor for a certain period of time ranges from 2 ms to maximum of 30 ms. We should also take into consideration that the

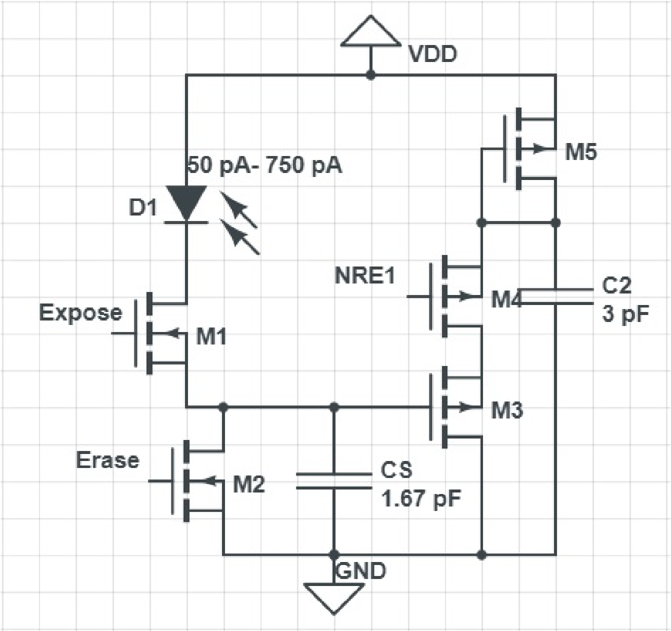


Figure 7. Top Level Analog Circuit Schematic for one pixel

corner cases are the lowest current driven for maximum exposure time and vice versa.

We designed the capacitor to give us an appropriate output voltage that should always maintain the buffer state to be in Active region. First, we discuss the buffer dc analysis to get the least voltage that should keep it on. We assume the load to be always in the perfect saturation region, so we expect the load to be V_{tn}. The voltage at the source is define in eq. (10)

$$V_{sg} = V_{dd} - V_{tn} \wedge V_{sg} > V_{tn} \tag{10}$$

As we should have V_{sg} at least V_{tn}, so that the buffer is always on. We get the following condition on the gate voltage in eq. (11)

$$V_g = V_s - V_{tn} = V_{dd} - 2V_{tn}. \tag{11}$$

Finally, we use the corner cases with the required voltage, to get the capacitance value in eq. (12)

$$C = \frac{I.T}{V_g} = \frac{I.T}{V_{dd} - 2V_{tn}} \tag{12}$$

2.5 Simulations and tests

This section consists of simulations of analog circuits using corner cases.

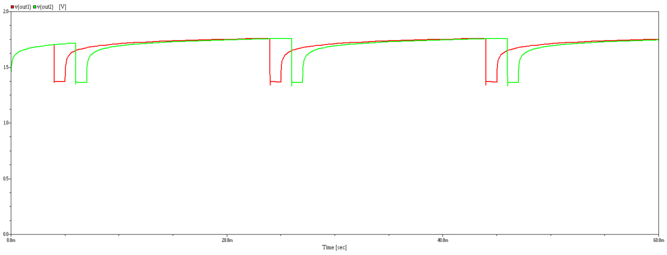


Figure 8. Output voltage of the 2 pixel circuits : current of 750 pA, exposure time 2ms

Photo-diode produces current of 750 pA, exposure signal frequency is 0.5 kHz, as a result we can observe these outputs from the 2 analog circuits.

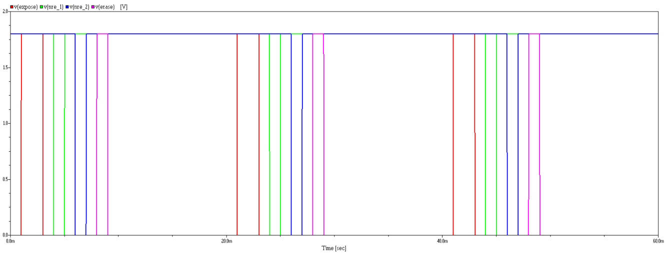


Figure 9. Input signals for the 2 pixel circuits : current of 750 pA, exposure time 2ms

Input signals Expose, Erase, NRE_1, NRE_2 with different timing delays to the analog circuits.

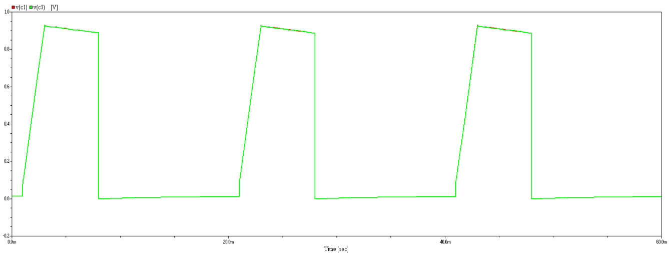


Figure 10. Output voltages of the 2 internal capacitors : current of 750 pA, exposure time 2ms

Internal capacitors are charged with current of 750 pA; output voltages are overlapped on this plot.

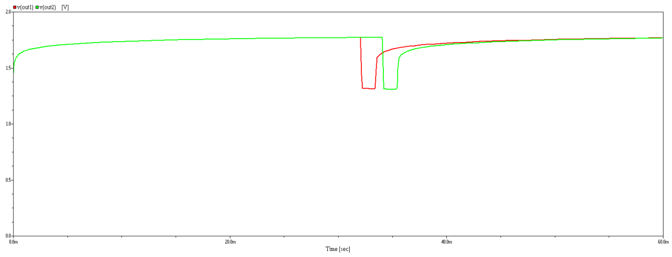


Figure 11. Output voltage of the 2 different pixel circuits : current of 50 pA, exposure time 30ms

Photo-diode produces current of 50 pA, exposure signal frequency is 33 kHz, as a result we can observe these outputs from the 2 analog circuits.

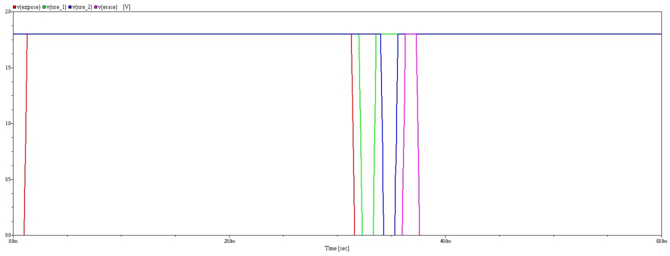


Figure 12. Input signals for the 2 pixel circuits : current of 50 pA, exposure time 30ms

Input signals Expose, Erase, NRE_1, NRE_2 with different timing delays to the analog circuits.

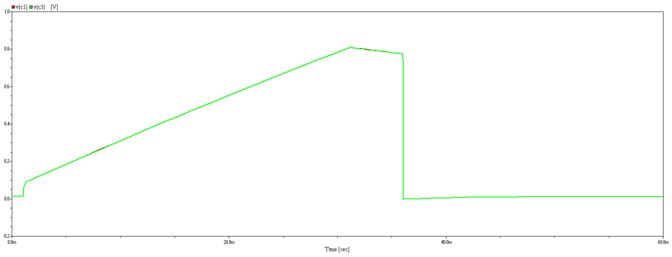


Figure 13. Output voltages of the 2 internal capacitors : current of 50 pA, exposure time 30ms

Internal capacitors are charged with current of 50 pA; output voltages are overlapped on this plot.

Now, we sum up all the design choices we made during our analog design phase:

3 Digital Part

3.1 Design Requirements

The digital part we are concerned about in this circuit is the one driving the control logic for the switches in the analog circuit.

3.1.1 Input and Output Logic

As we discussed in the earlier chapter; we can conclude the control logic in the following signals:

- $Init$: the signal indicating the start of the operation of the digital camera acts as the on switch for the whole system.
- $expose$: the expose signal controls the amount of time the current from photo-diode should be driven to the capacitor, as we said earlier, that should be maintained within the light conditions by the user, Here, we should test the highest and lowest conditions for this signal, and the convenience of it being maintained by the user.
- $erase$: As we seen earlier, we need to ensure that after every capture, an erase signal is driven to the NMOS parallel to the capacitance to discharge.
- nre_1, nre_2: the nre signals controls the switching between the 2 pixels and the adc input. Since it is controlled with a PMOS, it has to be designed as an active low signal.
- adc: the enable signal for the adc, it has to be enabled twice for each row of pixels to be running. the adc should be enabled for 1 clk cycle as stated by the specification.

3.1.2 Timing Chart

In this subsection, we are investigating the specified timing behaviour of the control signals. In **??**, we observe the timing for each signal, we use a clk frequency of 1 kHz. Starting with the "init" signal, its functionality is turning the core on, so it starts the process of taking a picture. Also, we should mention that the "init" signal is an input to the core as we will see in the block diagram in fig. 16. The "expose" signal should be high in the next cycle after "init" being high. Since specifications did not mention, the number of cycles "init" could be high, we maintain the functionality

Table 2
Summarizing Designing Choices in the Analog Circuit

Parameter	*Design Choice*	*Reason*
Expose, Erase, NRE switches Dim.	$\frac{W}{L}$= 1, W=L= 2μ	We had a trade off between speed and leakage current, due to the severe limitation on the driven current (50 pA), we leverage the leakage current over the speed, also due to the relaxed constrain for speed (1 MHz) on the other side.We chose the least $\frac{W}{L}$ within constrain.
Active Load Dim.	$\frac{W}{L}$= 1, W=L= 2μ	As shown in the previous chapters, we deduced an approximation expression for the gain equation eq. (8), from which we deduced that the Active load trans-conductance should be as small as possible with relative to the buffer trans-conductance eq. (9), we again chose the least possible $\frac{W}{L}$ within limits.
Buffer PMOS Dim.	$\frac{W}{L}$= 7, W= 10μ , L= 0.7μ	Here, we again based on the analysis in eq. (8), we chose the highest trans-conductance for the buffer circuit.
Sampling Capacitance Value	Cs=1.67pf	The capacitance is driven from the analysis in eq. (12), we chose the capacitance value that fits the worst case and result in a voltage enough to turn the buffer on.

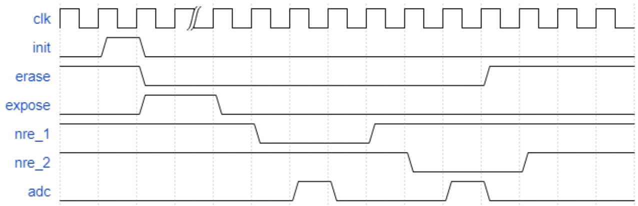


Figure 14. Timing Chart as indicated by specifications

regardless of how long it stays high, as long as it will be down before the cycle ends. so that we can detect the start of next cycle. The ”erase” signal should be up as long as there is no cycle or we are in the ”idle” state as we will see in the fig. 15. Finally the ”nre” signals are high for 3 consecutive cycles after exposure time is finished, ”adc” is enabled in the middle cycle for both ”nre” signals. we should also remark the buffer cycles between the exposure falling edge and the ”nre” rising edge, same between the 2 ”nre” signals.

3.2 Design Specifications

3.2.1 Finite State Machine

The FSM in fig. 15 just describes the timing diagram, we discussed in the previous chapters and also in the intro.. It is implemented using Verilog, the code is in section 5.

.

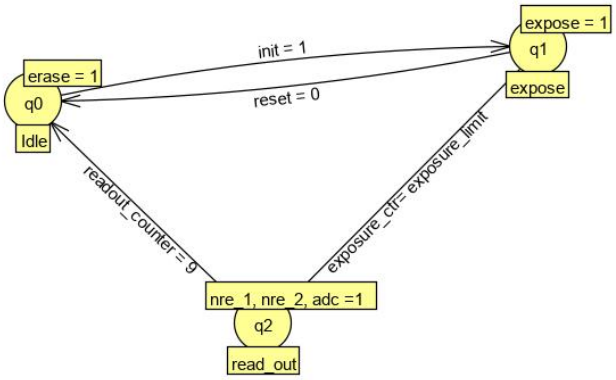


Figure 15. Moore FSM designed in section 5

3.2.2 Block Diagram

In fig. 16, we describes the block diagram we implemented in section 5. The block consists of 2 processes running in parallel, one is handling the exposure timing limit increment and decrement logic. Since the logic should be independent from the functionality of the digital camera. we implemented in a separate synchronous process. The ″exp_inc″ and ″exp_dec″ functionality should be available at any time regardless the finite state machine state, as long as it is within the pre-definied limits. The new limit would be preserved for the next cycle. For the FSM logic, it the same as fig. 15.

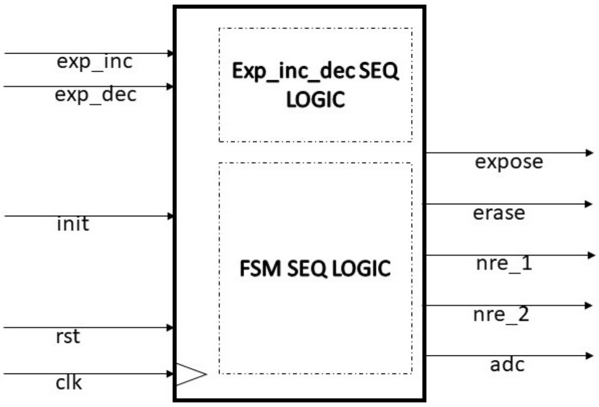


Figure 16. Block Diagram in section 5

3.3 Simulation and Testing

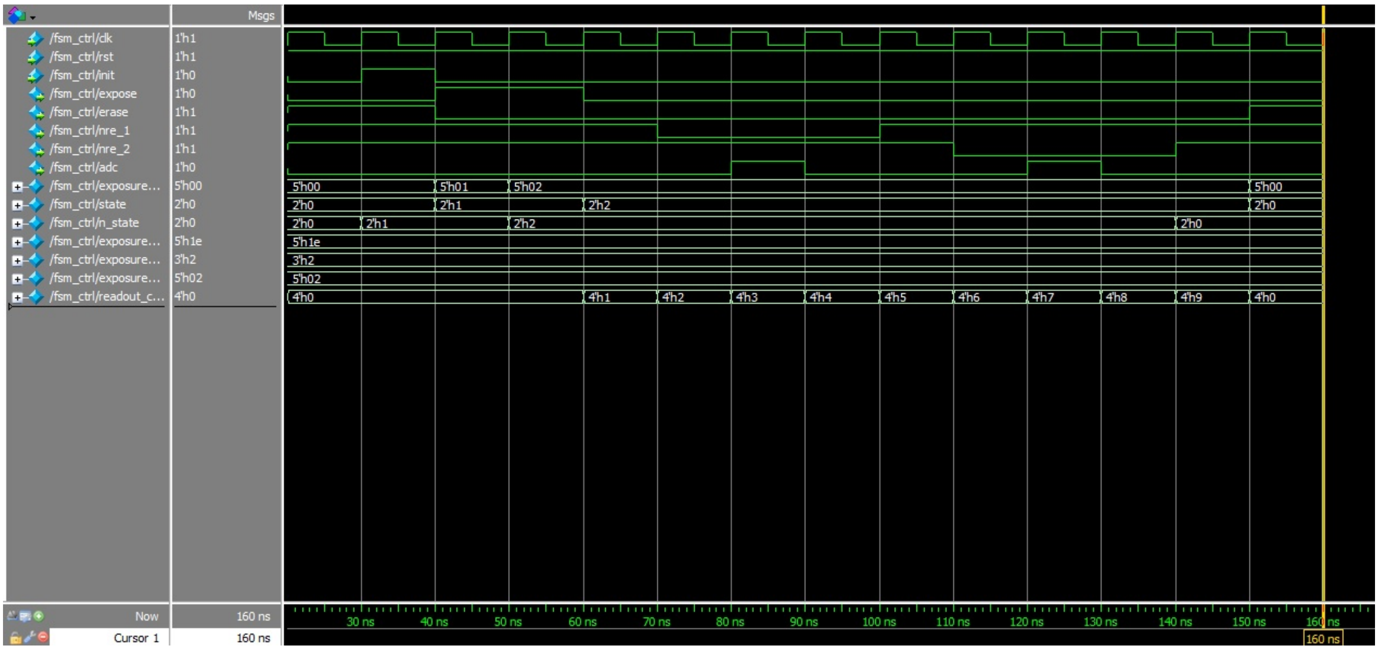


Figure 17. Simulating the basic functionality

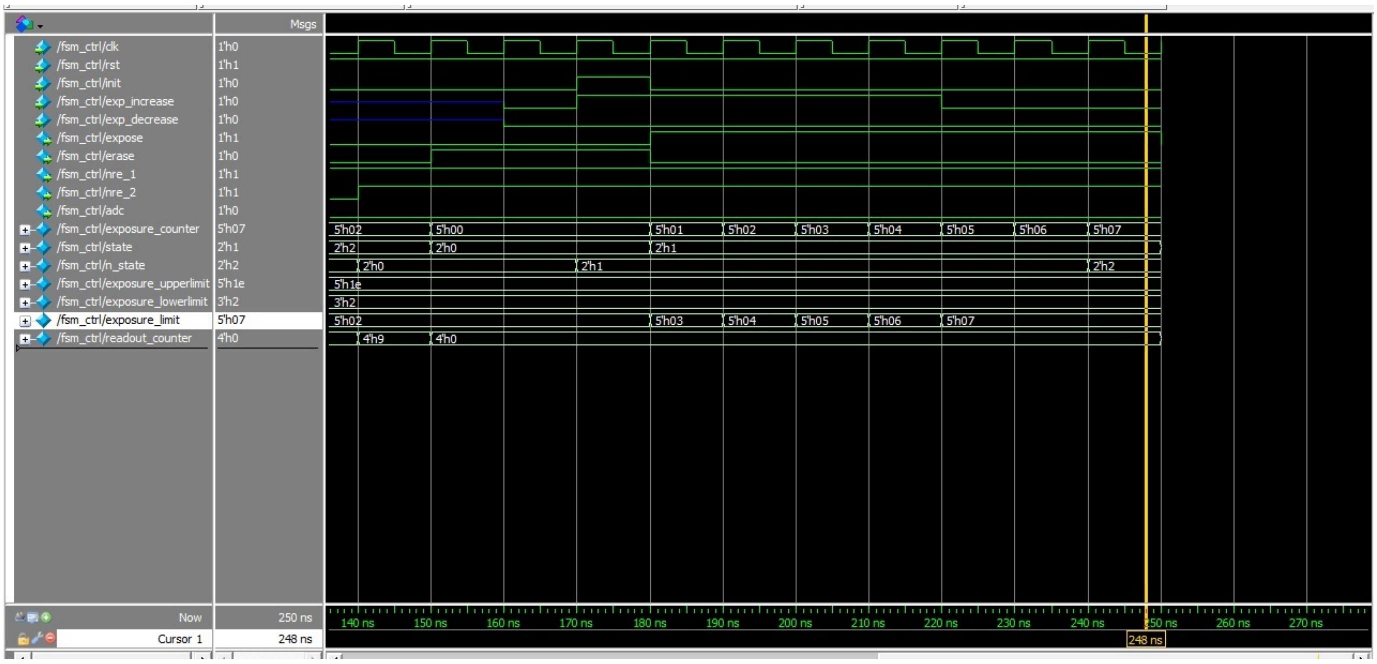
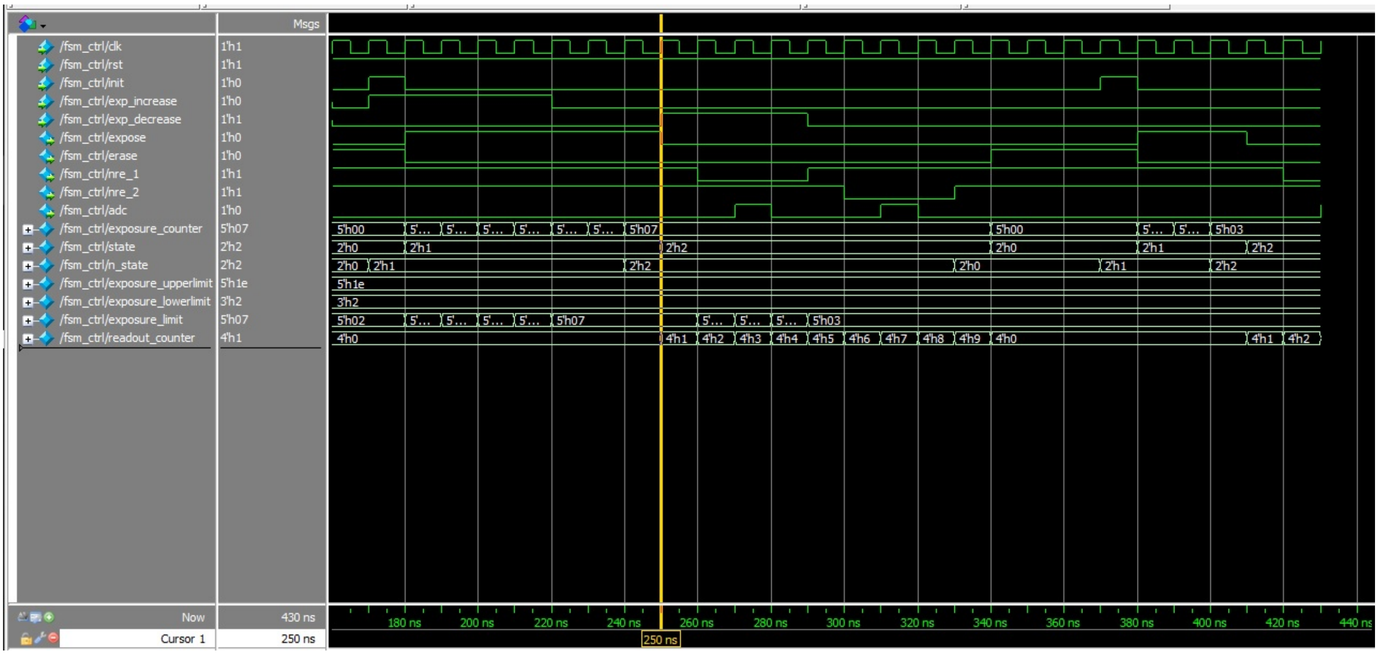

Msgs
/fsm_ctrl/clk 1'h0
/fsm_ctrl/rst 1'h1
/fsm_ctrl/init 1'h0
/fsm_ctrl/exp_increase 1'h0
/fsm_ctrl/exp_decrease 1'h0
/fsm_ctrl/expose 1'h1
/fsm_ctrl/erase 1'h0
/fsm_ctrl/nre_1 1'h1
/fsm_ctrl/nre_2 1'h1
/fsm_ctrl/adc 1'h0
/fsm_ctrl/exposure_counter 5'h07
/fsm_ctrl/state 2'h1
/fsm_ctrl/n_state 2'h2
/fsm_ctrl/exposure_upperlimit 5'h1e
/fsm_ctrl/exposure_lowerlimit 3'h2
/fsm_ctrl/exposure_limit 5'h07
/fsm_ctrl/readout_counter 4'h0
5'h02 5'h00 5'h01 5'h02 5'h03 5'h04 5'h05 5'h06 5'h07
2'h2 2'h0 2'h1
2'h0 2'h1 2'h2
5'h1e
3'h2
5'h02 5'h03 5'h04 5'h05 5'h06 5'h07
4'h9 4'h0
Now 250 ns
Cursor 1 248 ns
140 ns 150 ns 160 ns 170 ns 180 ns 190 ns 200 ns 210 ns 220 ns 230 ns 240 ns 250 ns 260 ns 270 ns
248 ns

Figure 18. Simulating the expose increase functionality



	Msgs
/fsm_ctrl/clk	1'h1
/fsm_ctrl/rst	1'h1
/fsm_ctrl/init	1'h0
/fsm_ctrl/exp_increase	1'h0
/fsm_ctrl/exp_decrease	1'h1
/fsm_ctrl/expose	1'h0
/fsm_ctrl/erase	1'h0
/fsm_ctrl/nre_1	1'h1
/fsm_ctrl/nre_2	1'h1
/fsm_ctrl/adc	1'h0
/fsm_ctrl/exposure_counter	5'h07
/fsm_ctrl/state	2'h2
/fsm_ctrl/n_state	2'h2
/fsm_ctrl/exposure_upperlimit	5'h1e
/fsm_ctrl/exposure_lowerlimit	3'h2
/fsm_ctrl/exposure_limit	5'h07
/fsm_ctrl/readout_counter	4'h1
Now	430 ns
Cursor 1	250 ns

Figure 19. Simulating both expose inc and decrease functionality

4 APPENDIX I

```
.include p18_cmos_models_tt.inc
.include p18_model_card.inc

.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 ENABLE = 1
.param EXPOSE = 5
.param NRE_1 = 7
.param NRE_2 = 14
.param ERASE = 4
.param OUT1 = 8
.param OUT2 = 13

.param TRF = {EXPOSURETIME/100} ! Risetime and falltime of EXPOSURE and ERASE
.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


.subckt PhotoDiode VDD N1_R1C1
I1_R1C1 VDD N1_R1C1 DC 750p
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 PhotoDiode




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_1 0 dc 0 pulse(VDD 0 NRE_R1_DLY TRF TRF CLK_PERIOD PERIOD)
VNRE_R2 NRE_2 0 dc 0  pulse(VDD 0 NRE_R2_DLY TRF TRF CLK_PERIOD PERIOD)


X1 1 2 PhotoDiode
*expose
MN1 2 EXPOSE 3 0 NMOS L=2U W=2U
*erase
MN2 3 ERASE 0 0 NMOS L=2U W=2U
*buffer
```

```
MN3 0 3 6 1 PMOS L=0.7U W=9U
*enable
MN4 6 NRE_1 OUT1 1 PMOS L=2U W=2U
*load
MN5 OUT1 OUT1 1 1 PMOS L=2U W=2U
*voltage cap
C1 3 0 1.67pF
C2 OUT1 0 3pf


 X2 1 10 PhotoDiode
*expose
MN6 10 EXPOSE 11 0 NMOS L=2U W=2U
*erase
MN7 11 ERASE 0 0 NMOS L=2U W=2U
*buffer
MN8 0 11 12 1 PMOS L=0.7U W=9U
*enable
MN9 12 NRE_2 OUT2 1 PMOS L=2U W=2U
*load
MN10 OUT2 OUT2 1 1 PMOS L=2U W=2U
*voltage cap
C3 11 0 1.67pF
C4 OUT2 0 3pf




 .plot V(OUT1) V(OUT2)
 .plot V(EXPOSE) V(NRE_1) V(NRE_2) V(ERASE)
 .plot V(C1) V(C3)
 *.plot V(3) V(6)
 *.plot V(11) V(12)
 *.plot V(3)
 *.plot V(NRE_1)
 *.plot V(NRE_2)
 *.plot V(C3)
 *.plot V(C1)
 *.plot V(12)
 *.plot V(6)
```

5 Appendix II

Listing 1. Verilog Code for the design written in 1 top level block

```
//'timescale 1 ms/1ps
// Module Definition: the module describes the FSM states and control logic

// Do a test-plan for the different specifications and Time plan, vs the Time
// Also Assumptions

// FSM Ctrl Interface
module fsm_ctrl (
                input wire clk, // Global Clk
                input wire rst, // Global Rst

                input wire init, // Init Signal
                input wire exp_increase,  // Exp_increase
                input wire exp_decrease,  // Exp_decrease

                output reg expose, // Active High Expose signals to all pixe
                output reg erase,  // Active High Erase signals to all pixels
                output reg nre_1,  // Active low Enable output signals to fi
                output reg nre_2,  // Active low Enable output signals to Se
                output reg adc     // Active High ADC enable signal
               );

// Internal Signal Definitions
reg [4:0] exposure_counter;

reg [1:0] state,n_state;
parameter IDLE = 2'b00, EXPOSE = 2'b01, READ_OUT = 2'b10;

reg [4:0]exposure_upperlimit;
reg [2:0]exposure_lowerlimit;
reg [4:0]exposure_limit;

reg [3:0] readout_counter;

// Sequential Exposure Set Logic
always @(posedge clk, negedge rst)
begin
        if (~rst)
        begin
        exposure_counter    <= 0;
        exposure_limit      <= 2;
        exposure_upperlimit <= 30;
        exposure_lowerlimit <= 2;
        end
        else
        begin
        if (exp_increase && exposure_limit<exposure_upperlimit)
                        exposure_limit=exposure_limit+1;
        if (exp_decrease && exposure_limit>exposure_lowerlimit)
                        exposure_limit=exposure_limit -1;
        end
end

// FSM logic
// Combinatorial Logic
always @(*)
begin
        case (state)
        IDLE: if (init ==1) n_state <= EXPOSE;
        EXPOSE: if (exposure_counter==exposure_limit) n_state<= READ_OUT;
        READ_OUT: if (readout_counter == 9) n_state <= IDLE;
        endcase
end

// Sequential state Logic
always @(posedge clk, negedge rst)
begin
        if (~rst)
        state = IDLE;
        else
        state = n_state;
end

// Sequential Output Logic
always @(posedge clk, negedge rst)
begin
        if (~rst)
        begin
        n_state<=IDLE;
        end
        else
        case (state)
        IDLE: begin
        exposure_counter<=0;
        readout_counter<=0;
        expose <= 0 ;
        erase <= 1;
        adc <=0;
        nre_1<=1;
        nre_2<=1;
        end
        EXPOSE: begin erase <=0; exposure_counter<=exposure_counter+1; expose
        READ_OUT: begin
        expose <=0;
        erase  <=0;
        readout_counter=readout_counter+1;
        if (readout_counter >1 && readout_counter <5)
                nre_1 <=0;
        else
                nre_1 <=1;
        if (readout_counter >5 && readout_counter <9)
                nre_2=0;
        else
                nre_2=1;
        if (readout_counter == 3 || readout_counter == 7)
                adc <= 1;
        else
                adc <= 0;
        end
        endcase
end

endmodule
```