World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:6, No:11, 2012

Image Sensor Matrix High Speed Simulation

Z. Feng, V. Viswanathan, D. Navarro, and I. O'Connor

***Abstract*—**This paper presents a new high speed simulation methodology to solve the long simulation time problem of CMOS image sensor matrix. Generally, for integrating the pixel matrix in SOC and simulating the system performance, designers try to model the pixel in various modeling languages such as VHDL-AMS, SystemC or Matlab. We introduce a new alternative method based on spice model in cadence design platform to achieve accuracy and reduce simulation time. The simulation results indicate that the pixel output voltage maximum error is at 0.7812% and time consumption reduces from 2.2 days to 13 minutes achieving about 240X speed-up for the 256x256 pixel matrix.

***Keywords*—**CMOS image sensor, high speed simulation, image sensor matrix simulation.

Open Science Index, Electronics and Communication Engineering Vol:6, No:11, 2012 publications.waset.org/6766/pdf

I. Introduction

As reviewed by E. R. Fossum and M. Bigas in the papers [1] [2], owing to its linear output response of the incident light intensity, the 3T-APS is the popular and basic pixel architecture in image sensors. A spice model has been reported by T. Reiner [3] studied mainly on the capacitance on the sensitive node and pixel transfer function. A detailed analysis has been given by A.El Gamal [4] on sensor dynamic range, system SNR, and several methods for improving the DR. The APS and PPS pixel has been modeled in VHDL-AMS in order to predict the chip behavior before the fabrication [5], but we can find that this model is not able to be as accurate as the spice model and did not take the noise and parasitic capacitor into account. Another APS VHDL model take the sense node nonlinearity into consideration and with the other functional block, but it suffers from internal errors when the imager matrix size becomes too big, the simulation time consumption is also an unavoidable problem [6]. The spice simulator based on numerical analysis, such as Spectre, Hspice, solves the circuit equation by iterative method, such as Newton's method, Newton-Raphson method. To be specific, for the imager resolution 256x256, the tool has to solve equations with respect to 196608 mosfet transistors and 65536 photodiodes suppose that the pixel is a 3T-APS. For simulating the whole sensor matrix, the simulator has to solve 130K equations [10]. For solving this big time and calculation consumption problem, we propose a high level simulation in Cadence design platform to simulate the imager on matrix level and check the sensor matrix performance in early stages of design time.

We introduce the image sensor pixel models and basic signal processing procedure in the section II. In section III, we will discuss the fast simulation method and present a comparison result. Finally, in section IV we conclude and provide the future perspectives.

Z. Feng is with the Lyon Institute of Nanotechnology, Ecole Centrale de Lyon, Ecully, 69134 France, (phone: +33 0472186059; fax: +33 0478433593; e-mail: zhenfu.feng@ ec-lyon.fr).

V. Viswanathan is with the Lyon Institute of Nanotechnology, Ecole Centrale de Lyon, Ecully, 69134 France, (phone: +33 0472186059; fax: +33 0478433593; e-mail: Vijay.Viswam@ ec-lyon.fr).

D. Navarro is with the Lyon Institute of Nanotechnology, Ecole Centrale de Lyon, Ecully, 69134 France, (phone: +33 0472186398; fax: +33 0478433593; e-mail: David.Navarro@ ec-lyon.fr).

I. O'Connor is with the Lyon Institute of Nanotechnology, Ecole Centrale de Lyon, Ecully, 69134 France, (phone: +33 0472186054; fax: +33 0478433593; e-mail: Ian.Oconnor@ ec-lyon.fr).

II. Image Sensor Model and Signal Analysis

As shown in Fig 1, the 3T-APS consists of three NMOS transistors and a photodiode. Typical signal readout procedure can be roughly divided into three phases, such as reset, integration and readout. During the reset operation, the sense node full well capacitor C_{pd} which comprise the photodiode inner capacitor, reset transistor MN1 source capacitor, and the MN2 gate parasitic capacitor will be reset to a voltage V_{pd}. The charges will be accumulated on the sense nodes and it can gain maximum charge Q_{max} which depends on the full well capacitance of the photodiode. After resetting, the capacitor C_{pd} will be discharged by the photocurrent (I_{ph}) and dark current (I_{dark}), the former is proportional to the light intensity during the integration time and I_{dark} is the leakage current flowing through the photodiode when no photons enter the image pixel, the main part of the total dark current is coming from the depletion of the photodiode edge at the surface [7]. As shown in Fig 2, the slope of discharge curve is determined by the sum of I_{ph} and I_{dark} and it also determine by the C_{pd}. The bigger the I_{ph} is, the faster the C_{pd} discharges. For example, using three different light intensities as input to the simulation produces three different I_{ph} giving three discharge curves with different slopes, as shown in Fig 2. Meanwhile, the discharge voltage is buffered by inner source follower MN2. The access transistor MN3 passes this voltage to column bus according to the readout timing. The column voltage will be sampled twice by the Correlated Double Sampling (CDS) circuit, which is used to reduce reset noise and fixed pattern noise [8]. The first sample happens in the reset period and the second sample happens during integration. The final output voltage can be expressed by equation (1):

$$V_{OS} = V_{CDS-H} - V_{CDS-S} \quad (1)$$

where, V_{CDS_H} is the output voltage in reset phase and V_{CDS_S} is the output voltage sampled during the integration. The differential voltage V_{os} is the final signal passed to Analog to Digital converter (ADC) for generating the image numeric data.

Open Science Index, Electronics and Communication Engineering Vol:6, No:11, 2012 publications.waset.org/6766/pdf

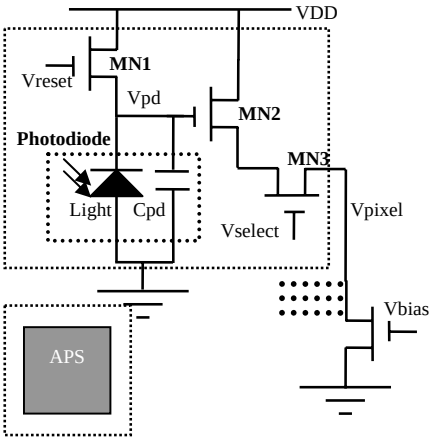


Fig. 1 3T-APS pixel model

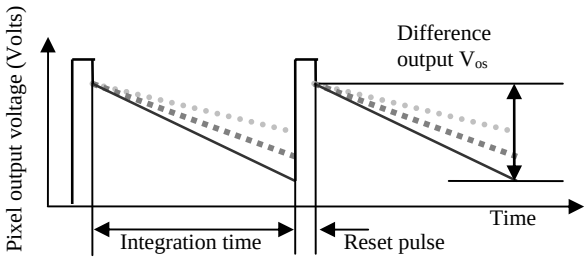


Fig. 2 Timing diagram of 3T-APS [3]

III. High Speed Simulation Method and Implementation

The spice simulator is commonly used to simulate the circuit behavior. Imager matrix is built up from pixel blocks that are repeated thousands or millions of times. The number of the circuit element in such a matrix is too huge for the common spice simulator. The reason why the Spectre simulator suffers from memory and runtime problem for huge pixel matrix is that the nets and terminals in the matrix form a very complicated equation with lots of variables according to Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL) principle. It costs lot of time to solve the equations and it consumes lot of memory to execute the calculation. Currently designers simulate pixel matrix in a very limited size, such as 10x10 or 20x20 before the fabrication owing to these difficulties. In order to give high level performance estimation, we present our high level and high speed approach. The main idea in this work is to use one pixel instead of the whole matrix in the simulation with the help of the generated database to map the input photocurrent and output voltage. We use 3T-APS as a standard cell as shown in the Fig 3, in order to obtain an correlated output voltage database V[i][j] we perform a parametric simulation with the photocurrent parameter $I_{ph}[i][j]$ which is the input signal of the pixel with address row i and column j in the pixel matrix. In this way, we convert the photocurrent values to its correlated output voltage database and we form a Look Up Table (LUT) by these data shown as the blocks marked $I_{ph}[i][j]$ and V[i][j] in Fig 3.

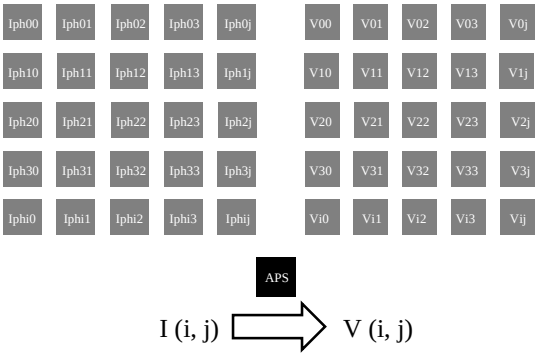


Fig. 3 Mapping and generated database

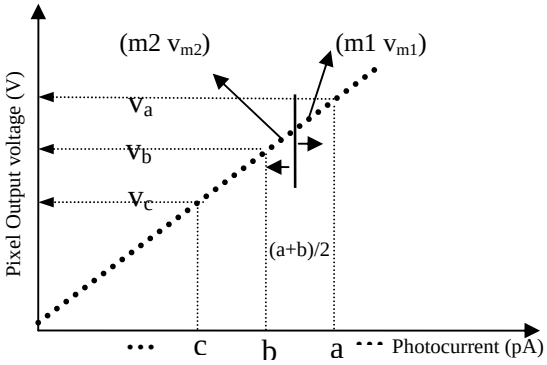


Fig. 4 Pixel output voltage versus photocurrent

An 8 bit depth grey level image with intensity data ranging from 0 to 255 is used to represent the output voltages. So we need 256 different output voltage stages from the lowest to the highest corresponding to numeric intensity data from 0 to 255 and these voltage stages all relate to a specific photocurrent. The pixel simulation output voltage increasing linearly with the light intensity is shown in Fig 4. The output voltage curve implies that some pixel output voltage can be estimated before the simulation. Since we need 256 different output voltage stages, we divide the input photocurrents into 256 blocks and in turns we get a series of mean photocurrent of each block. We create an input table of these 256 mean values and we apply these 256 values in the simulation. Another table of 256 output voltages is generated after the simulation as shown in table I. And then, a mapping function is used to match and set each pixel in the pixel matrix output voltage directly.

TABLE I
THE PHOTOCURRENT AND OUTPUT VOLTAGE

Photocurrent (A)	Output voltage (Volts)
a: [Iph_00]	V_a:V_{00}
b: [Iph_01]	V_b:V_{01}
c: [Iph_02]	V_c:V_{02}
…	…
x	y

We assume that the photocurrent is a random unknown value as x and there is a threshold photocurrent in each range such as (a+b)/2 in region [a b]. As shown in Fig 4, since the

Open Science Index, Electronics and Communication Engineering Vol:6, No:11, 2012 publications.waset.org/6766/pdf

point m1 has a bigger photocurrent value than the threshold (a+b)/2, so the correlated output CDS voltage V_{m1} is closer to V_a than V_b according to the circuit behavior presented by dashed line which stands for the output voltage increasing linearly along with the photocurrent. Every output voltage corresponds to an intensity data value in the grey level image, for example, V_a is converted to 165 in the numeric intensity data in the image and V_b to 164. Owing to the V_{m1} is closed to V_a and there is no available integer value between 164 and 165 in the image. So the voltage V_{m1} could be evaluated to V_a and Vm2 could be V_b if the simulation results is shown as an grey level image. So in this way as shown in Fig 3, every photocurrent I(i j) value in the scene will be matched in the generated database and every correlated output voltage V(i j) will be estimated.

We have implemented this LUT function in our own image sensor matrix simulation tool box shown in Fig 5. This tool can help to investigate how the pixel performance affects the output image quality [9]. Two levels simulation are available for checking the pixel performance, user can investigate the single pixel character by performing the normal simulation and he can also perform the matrix level simulation which can be used to investigate the high level performance. For example, user can investigate the affection of pixel temperature changes on the final output image intensity data. Further more, the column capacitance has been taken into account in the big matrix simulation. With the help of this high speed approach, 130K equations needed to be solved by spice simulator for simulating the pixel matrix 256x256 have been reduced to 4.8K in our new approach and simulation time has been greatly saved. The time performance is shown in Fig 6. The time consumption of Spectre classical simulation increases exponentially with matrix row numbers, whereas it increases slowly after the initialization which takes 12 minutes to form the LUT database in the new fast simulation method.

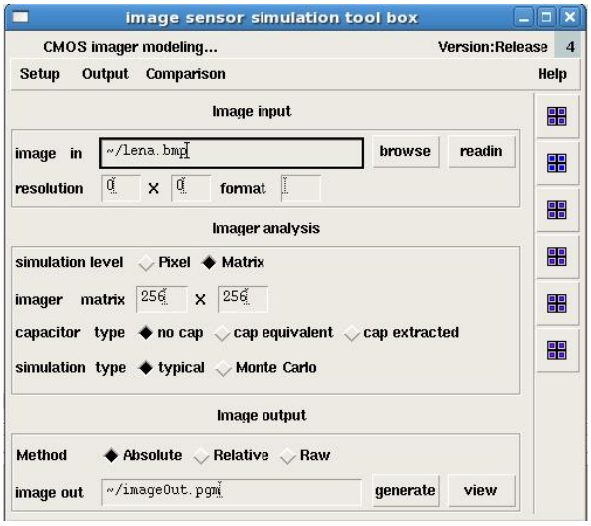


Fig. 5 Imager matrix simulation toolbox in Cadence ADE

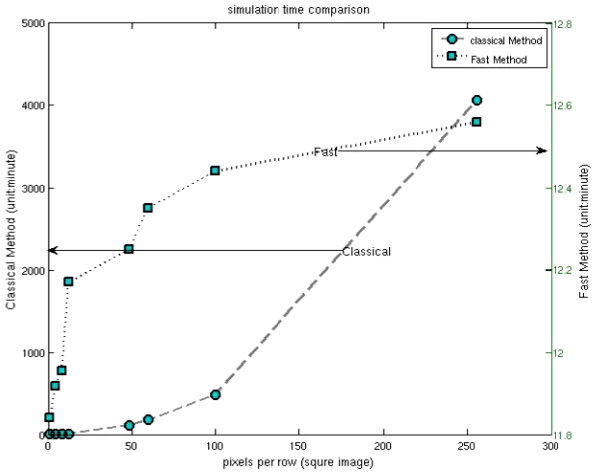


Fig. 6 Time consumption versus pixel numbers

Table II is used to show the time consumption comparison. For the matrix in small size, such as 4x4, 8x8 and 12x12 pixels, the classical simulation time consumption is 7 minutes, but the simulation time is 8 hours when the matrix size reach 100x100. In contrast, the initialization time for the fast simulation is around 12 minutes and the mapping function can be finished very quickly, so this fast simulation method is quite useful for the simulation when the matrix size is bigger than 12x12. The mapping function consumes about 1 second to map the whole matrix output of size 1024x768.

TABLE II
TIME CONSUMPTION COMPARISON

Matrix size	8 x8	12 x12	60 x60	256 x256	512 x512
Classical (min)	3.7	7.5	179.23	3465	error
Initialization(min)	11.57	12.01	12.21	12.3	12.31
Mapping(ms)	23.6	34.6	39.5	91.4	483

As it is a high level estimation method, validation of the results with a reference simulation is performed. The validation is based on AMS 0.35um technology and 3T-APS model with the fixed integration time. It is performed by simulating 256x256 classical simulations. The following image generation method [9] is used to convert the output voltage into gray level images.

$$grey_relative = 255 \times \frac{V_{os}}{V_{\max} - V_{\min}} \quad (2)$$

$$grey_raw = 255 \times \frac{V_{os}}{V_{CC}} \quad (3)$$

$$grey_abs = 255 \times \frac{V_{os}}{V_{rst}} \quad (4)$$

where, the V_{os} is the CDS output voltage, V_{max} and V_{min} is the maximum and minimum value of the pixel matrix V_{os} values, respectively. The results of classical Spectre simulation and the new approach are shown in Fig 7.

Open Science Index, Electronics and Communication Engineering Vol:6, No:11, 2012 publications.waset.org/6766/pdf



Fig. 7 Top-left is the original input picture, Top-right is the grey file, bottom-left is the classical simulation result, and bottom-right is the fast simulation result

The original input and output picture have a resolution 256x256 and a wider dynamic range in output images has been achieved owing to the relative image generation method. The histogram comparison is shown in Table III.

TABLE III
IMAGE HISTOGRAM COMPARISON

	mean	std	median
original	125.7	60.5	138
grey	89.4	45.5	92
classical	139.7	79.4	153
fast	139.6	79.8	154

The comparison shows that the mean value in classical simulation results is 139.7 and standard variation is 79.4, median value is 153, whereas the value is 139.6, 79.8 and 154, respectively in the fast simulation results. It proves that the fast simulation results match the classical results. We compare each output voltage with the fast simulation and find the maximum error is about 3.647506 uV. With the help of equation (5), (6) and (7), a point to point pixel output numeric data comparison has been used to measure the data intensity error.

$$err_{abs} = p_{fast} - p_{classical} \quad (5)$$

$$err_{rel} = \frac{err_{abs}}{256} \times 100\% \quad (6)$$

$$err_{mean} = \frac{\sum_{i}^{m}\sum_{j}^{n} err_{abs}}{m \times n} \times \frac{1}{256} \times 100\% \quad (7)$$

where, err_{abs}, err_{rel} and err_{mean} represent the absolute, relative and mean pixel intensity data error, respectively.

TABLE IV
SIMULATION RESULTS COMPARISON AND ERROR

Simulation	V_{reset}	V_{sample}	Raw	Rel	Abs
Classical	1.327557	0.4471916	68	161	169
Fast	1.327467	0.440786	68	161	171

The Table IV shows the worst case of the simulation results and correlated intensity data. V_{reset} is the output voltage at the reset stage which is equal to V_{CDS-H}. V_{sample} is the output voltage after integration which is equal to V_{CDS-S}. The maximum error of intensity data is 2 and it comes from the absolute image generation method shown in equation (4), using this equation does not correct the error in V_{reset}. It is about 0.78% (2/256) of data range 256 in grey level. The mean value which stands for the whole matrix simulation average error in intensity data is quite small and is about 0.1681% of the output data range.

IV. Conclusion and Discussion

This paper presents an alternative way of image sensor matrix simulation. This method is technology independent and the accurate spice model could be used to simulate the image sensor matrix instead of using a high level language model, such as VHDL-AMS model. The result has proved strongly that our new fast simulation approach can be used for accelerating the CMOS image sensor pixel matrix simulation. Due to the time consumption has been reduced largely, this makes the CMOS image sensor simulation more efficient and in turn it is possible to check the system performance in a very short time. This new improvement makes realizing the whole image generation and processing image data possible in Cadence. More cases should be simulated to verify and prove this high level and high speed simulation approach. The future work will be focused on the aspects such as: variability, technologies and pixel architectures. The pixel matrix FPN will be investigated and analyzed.

References

[1] E. R. Fossum, “CMOS Image Sensor: Electronic Camera-on-A-chip”, IEEE Transactions On Electronic Devices, vol. 44, N.10, 1997.
[2] M.Bigas, “Review of CMOS image sensors” Microelectronics journal 37(2006) 433-451.
[3] T. Reiner, “CMOS Image Sensor 3T Nwell Photodiode Pixel Spice Model” 23rd IEEE Convention of Electrical and Electronics Engineers in Israel. pp 161-164, 6-7 Sept. 2004.
[4] A. El Gamal, “High dynamic range image sensors,” Tutorial at International Solid-State Circuits Conference, February 2002.
[5] F. Dadouche, A. Pinna, P. Garda, A. Alexandre. Modeling of Pixel Sensors for Image Systems with VHDL-AMS. Proc. IEEE DTIS, Tunisia, September 5-7, 2006. pp. 289-293.
[6] D. Navarro, D. Ramat, F. Mieyeville, I. O'Connor, F. Gaffiot, L. Carrel, "VHDL & VHDL-AMS modeling and simulation of a CMOS imager IP", Forum on specification & Design Languages, Lausanne, Switzerland, September 2005.
[7] N. V. Loukianova, “Leakage current modeling of test structures for characterization of dark current in CMOS image sensors,” IEEE Trans. Electron Dev., vol. 50, no. 1, pp. 77 - 83, 2003
[8] M. White, D. Lampe, F. Blaha, and I. Mack, “Characterization of surface channel CCD image arrays at low light levels” IEEE J. Solid-State Circuits, vol. SC-9, pp. 1–13, Sept. 1974.
[9] D. Navarro, Z. Feng, V. Viswanathan, L. Carrel, I. O’Connor, “Image toolbox for CMOS image sensors simulations in Cadence ADE”, DeMset2011 conference, Orlando, Florida, USA, Dec 2011.
[10] Cadence User Guide